Pacemaker 1.1

Clusters from Scratch

Creating Active/Passive and Active/Active Clusters on Fedora

Andrew Beekhof

Pacemaker 1.1 Clusters from Scratch
Creating Active/Passive and Active/Active Clusters on Fedora
Edition 3

Author Andrew Beekhof andrew@beekhof.net

Copyright © 2010 Andrew Beekhof This material may only be distributed subject to the terms and conditions set forth in the GNU Free Documentation License (GFDL), V1.2 or later (the latest version is presently available at *http://www.gnu.org/licenses/fdl.txt*).

The purpose of this document is to provide a start-to-finish guide to building an example active/passive cluster with Pacemaker and show how it can be converted to an active/active one.

The example cluster will use:

1. Fedora 13 as the host operating system
2. Corosync to provide messaging and membership services,
3. Pacemaker to perform resource management,
4. DRBD as a cost-effective alternative to shared storage,
5. GFS2 as the cluster filesystem (in active/active mode)
6. The crm shell for displaying the configuration and making changes

Given the graphical nature of the Fedora install process, a number of screenshots are included. However the guide is primarily composed of commands, the reasons for executing them and their expected outputs.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F1`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F7`** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

[1] https://fedorahosted.org/liberation-fonts/

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' won't cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

You should over ride this by creating your own local Feedback.xml file.

Read-Me-First

1.1. The Scope of this Document

The purpose of this document is to definitively explain the concepts used to configure Pacemaker. To achieve this best, it will focus exclusively on the XML syntax used to configure the CIB.

For those that are allergic to XML, Pacemaker comes with a cluster shell and a Python based GUI exists, however these tools will not be covered at all in this document [1] , precisely because they hide the XML.

Additionally, this document is NOT a step-by-step how-to guide for configuring a specific clustering scenario. Although such guides exist, the purpose of this document is to provide an understanding of the building blocks that can be used to construct any type of Pacemaker cluster.

1.2. What Is Pacemaker?

Pacemaker is a cluster resource manager. It achieves maximum availability for your cluster services (aka. resources) by detecting and recovering from node and resource-level failures by making use of the messaging and membership capabilities provided by your preferred cluster infrastructure (either *Corosync*[2] or Heartbeat).

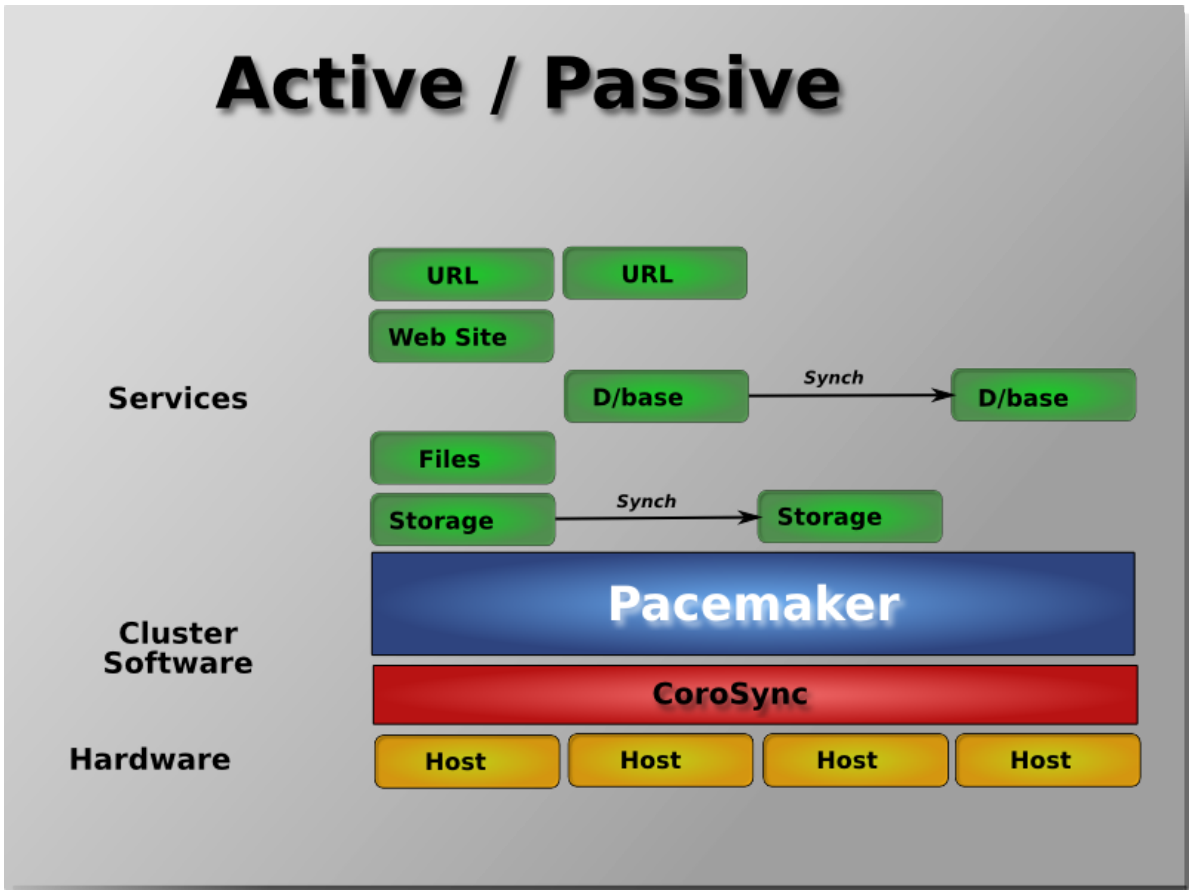
Pacemaker's key features include:

- Detection and recovery of node and service-level failures
- Storage agnostic, no requirement for shared storage
- Resource agnostic, anything that can be scripted can be clustered
- Supports *STONITH*[3] for ensuring data integrity
- Supports large and small clusters
- Supports both *quorate*[4] and *resource driven*[5] clusters
- Supports practically any *redundancy configuration*[6]
- Automatically replicated configuration that can be updated from any node
- Ability to specify cluster-wide service ordering, colocation and anti-colocation
- Support for advanced services type
  - Clones: for services which need to be active on multiple nodes
  - Multi-state: for services with multiple modes (eg. master/slave, primary/secondary)
- Unified, scriptable, cluster shell

[1] It is hoped however, that having understood the concepts explained here, that the functionality of these tools will also be more readily understood.
[2] http://www.corosync.org/

1.3. Types of Pacemaker Clusters

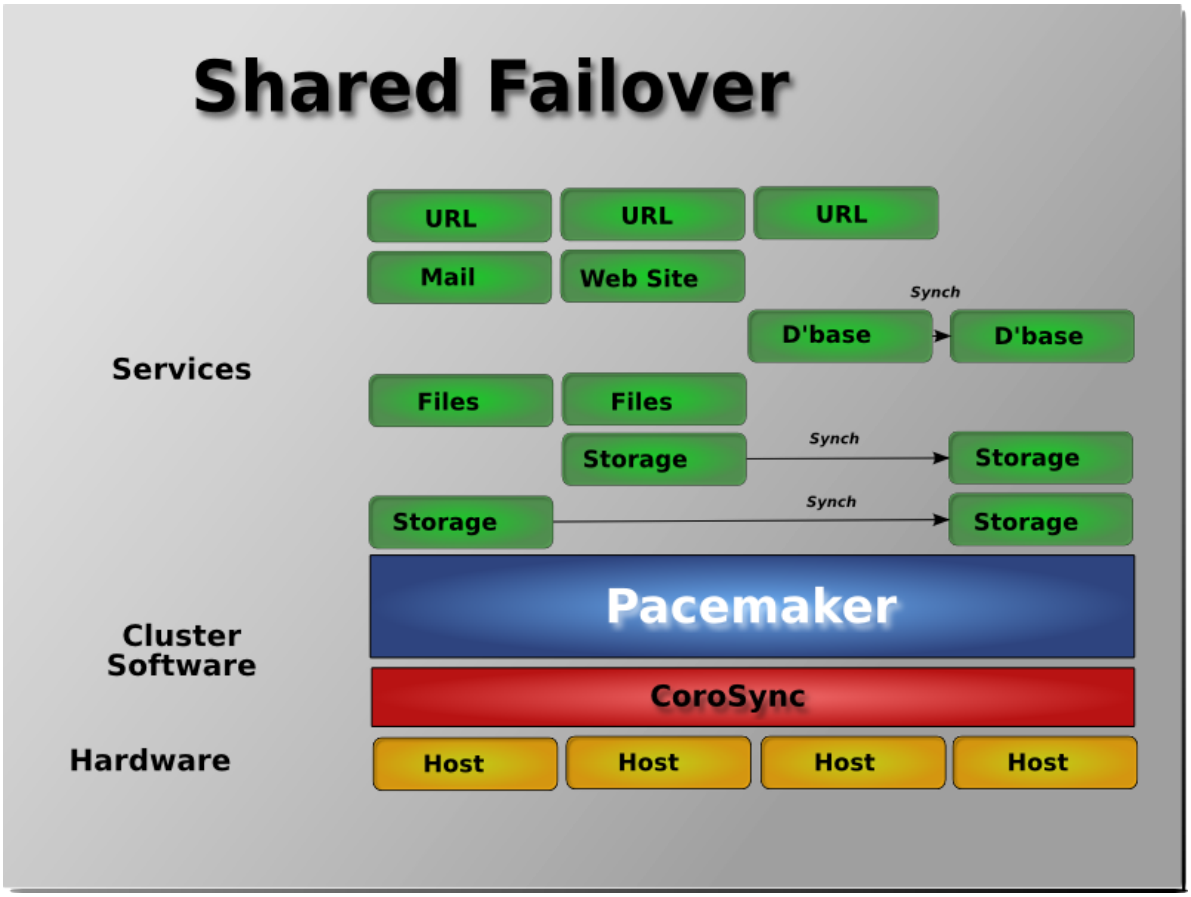
Pacemaker makes no assumptions about your environment, this allows it to support practically any *redundancy configuration*[7] including Active/Active, Active/Passive, N+1, N+M, N-to-1 and N-to-N.



Two-node Active/Passive clusters using Pacemaker and DRBD are a cost-effective solution for many High Availability situations.

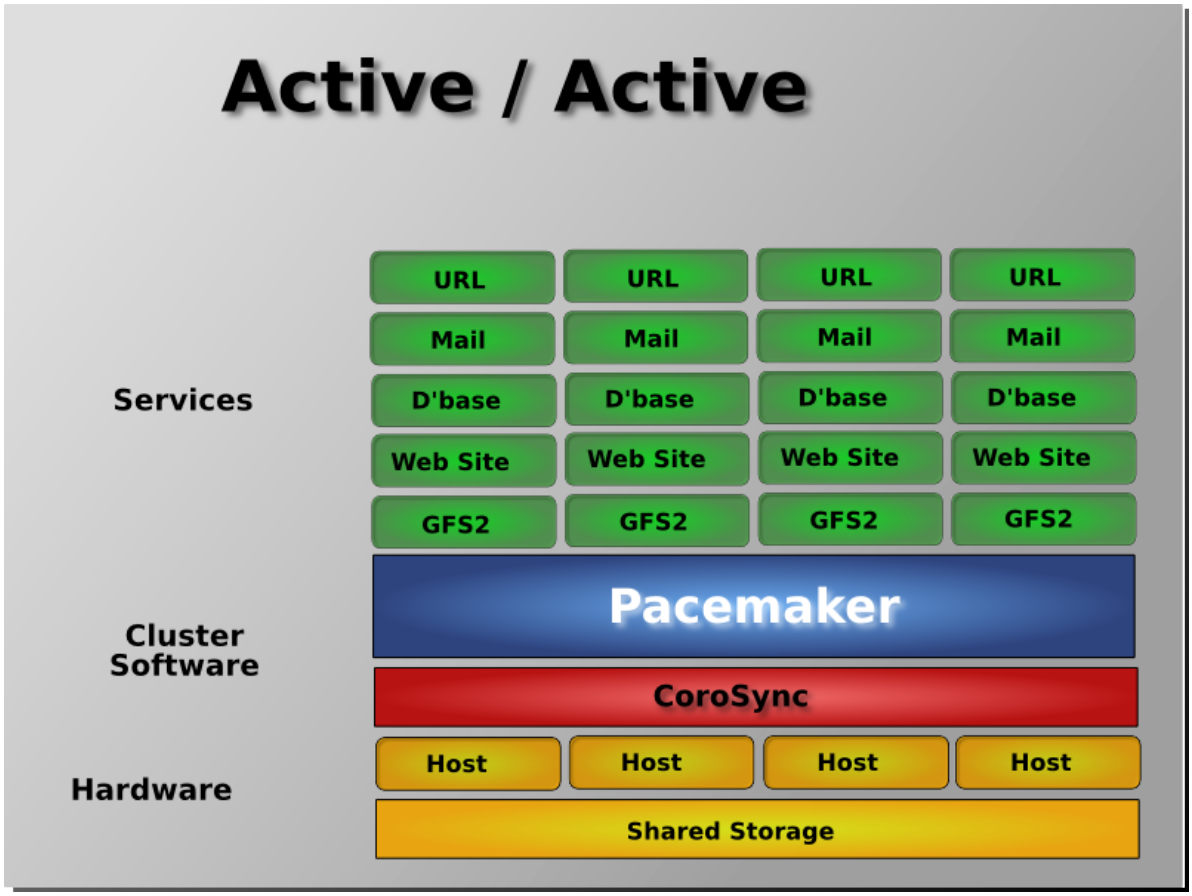
Figure 1.1. Active/Passive Redundancy

[7] http://en.wikipedia.org/wiki/High-availability_cluster#Node_configurations



By supporting many nodes, Pacemaker can dramatically reduce hardware costs by allowing several active/passive clusters to be combined and share a common backup node

Figure 1.2. Shared Failover



When shared storage is available, every node can potentially be used for failover. Pacemaker can even run multiple copies of services to spread out the workload.

Figure 1.3. N to N Redundancy

1.4. Pacemaker Architecture

At the highest level, the cluster is made up of three pieces:

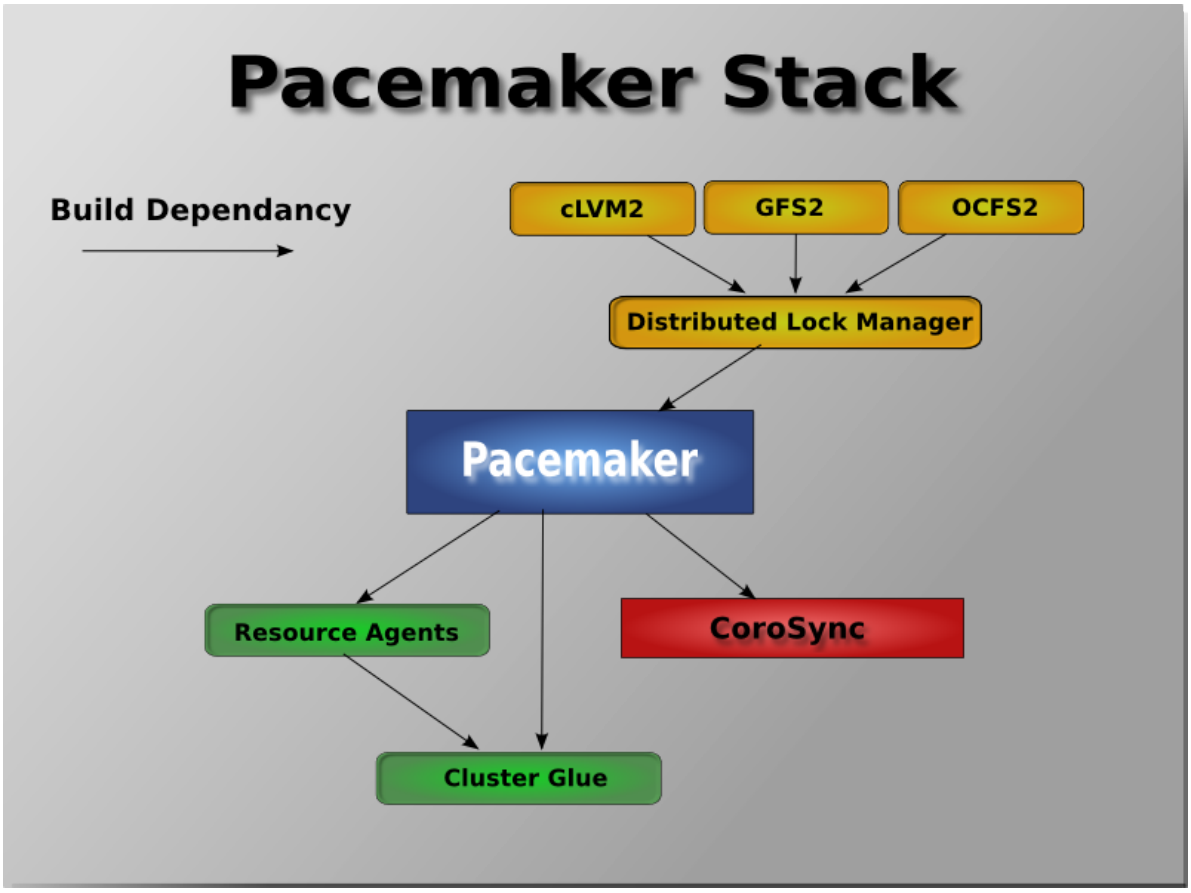
- Core cluster infrastructure providing messaging and membership functionality (illustrated in red)
- Non-cluster aware components (illustrated in blue). In a Pacemaker cluster, these pieces include not only the scripts that knows how to start, stop and monitor resources, but also a local daemon that masks the differences between the different standards these scripts implement.
- A brain (illustrated in green) that processes and reacts to events from the cluster (nodes leaving or joining) and resources (eg. monitor failures) as well as configuration changes from the administrator. In response to all of these events, Pacemaker will compute the ideal state of the cluster and plot a path to achieve it. This may include moving resources, stopping nodes and even forcing them offline with remote power switches.

Conceptual overview of the cluster stack

Figure 1.4. Conceptual Stack Overview

When combined with Corosync, Pacemaker also supports popular open source cluster filesystems [8] Due to recent standardization within the cluster filesystem community, they make use of a common distributed lock manager which makes use of Corosync for its messaging capabilities and Pacemaker for its membership (which nodes are up/down) and fencing services.

[8] Even though Pacemaker also supports Heartbeat, the filesystems need to use the stack for messaging and membership and Corosync seems to be what they're standardizing on. Technically it would be possible for them to support Heartbeat as well, however there seems little interest in this.



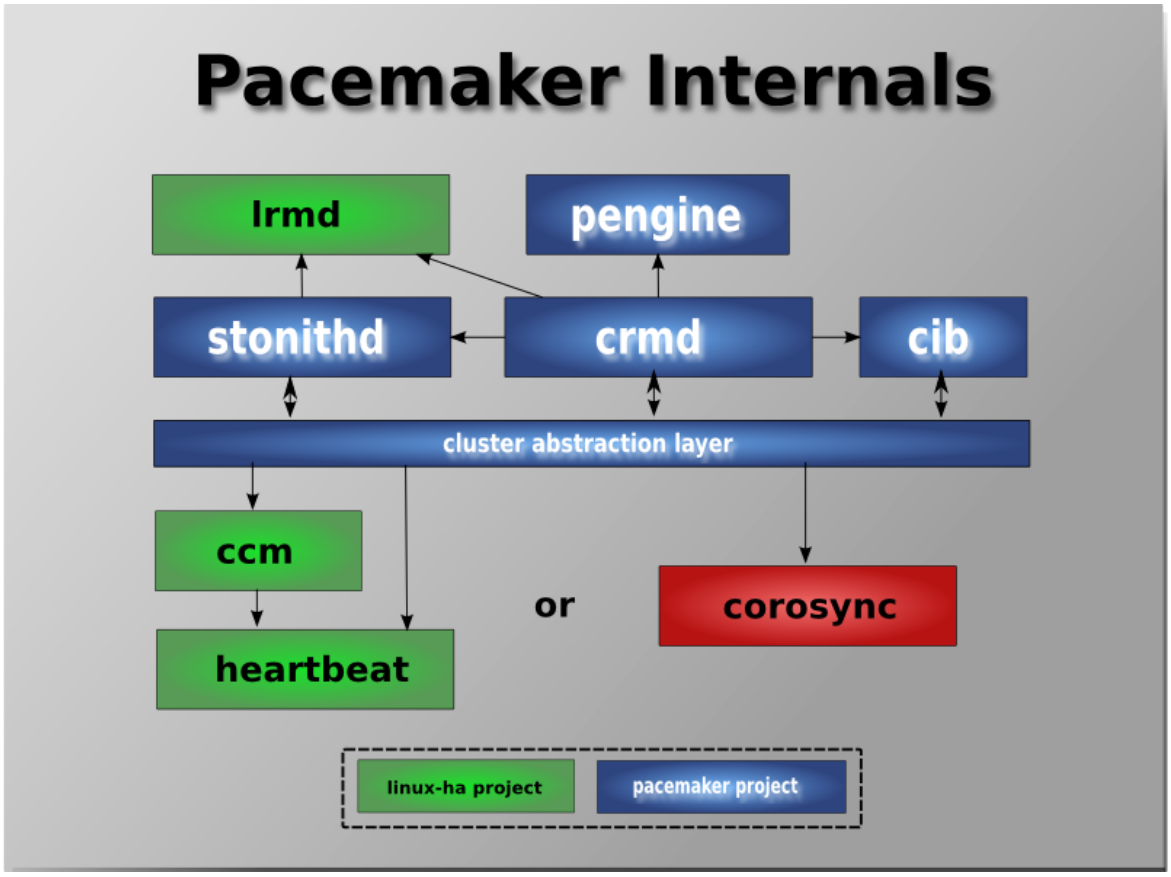
The Pacemaker stack when running on Corosync

Figure 1.5. The Pacemaker Stack

1.4.1. Internal Components

Pacemaker itself is composed of four key components (illustrated below in the same color scheme as the previous diagram):

- CIB (aka. Cluster Information Base)
- CRMd (aka. Cluster Resource Management daemon)
- PEngine (aka. PE or Policy Engine)
- STONITHd



Subsystems of a Pacemaker cluster running on Corosync

Figure 1.6. Internal Components

The CIB uses XML to represent both the cluster's configuration and current state of all resources in the cluster. The contents of the CIB are automatically kept in sync across the entire cluster and are used by the PEngine to compute the ideal state of the cluster and how it should be achieved.

This list of instructions is then fed to the DC (Designated Co-ordinator). Pacemaker centralizes all cluster decision making by electing one of the CRMd instances to act as a master. Should the elected CRMd process, or the node it is on, fail... a new one is quickly established.

The DC carries out the PEngine's instructions in the required order by passing them to either the LRMd (Local Resource Management daemon) or CRMd peers on other nodes via the cluster messaging infrastructure (which in turn passes them on to their LRMd process).

The peer nodes all report the results of their operations back to the DC and based on the expected and actual results, will either execute any actions that needed to wait for the previous one to complete, or abort processing and ask the PEngine to recalculate the ideal cluster state based on the unexpected results.

In some cases, it may be necessary to power off nodes in order to protect shared data or complete resource recovery. For this Pacemaker comes with STONITHd. STONITH is an acronym for Shoot-The-Other-Node-In-The-Head and is usually implemented with a remote power switch. In Pacemaker, STONITH devices are modeled as resources (and configured in the CIB) to enable them to be easily monitored for failure, however STONITHd takes care of understanding the STONITH topology such that its clients simply request a node be fenced and it does the rest.

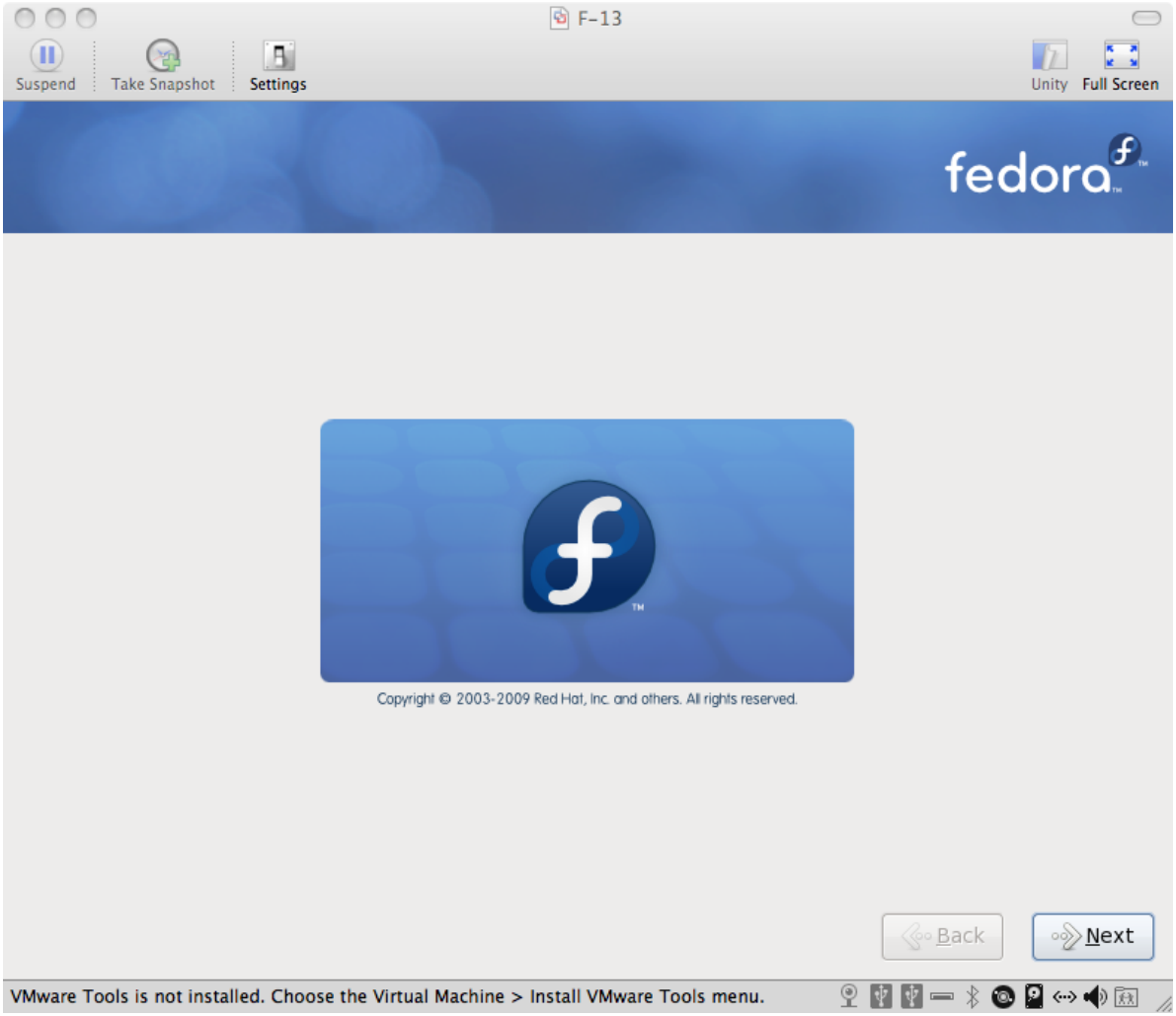
Installation

2.1. OS Installation

Detailed instructions for installing Fedora are available at *http://docs.fedoraproject.org/install-guide/f13/* in a number of languages. The abbreviated version is as follows...

Point your browser to *http://fedoraproject.org/en/get-fedora-all*, locate the *Install Media* section and download the install DVD that matches your hardware.

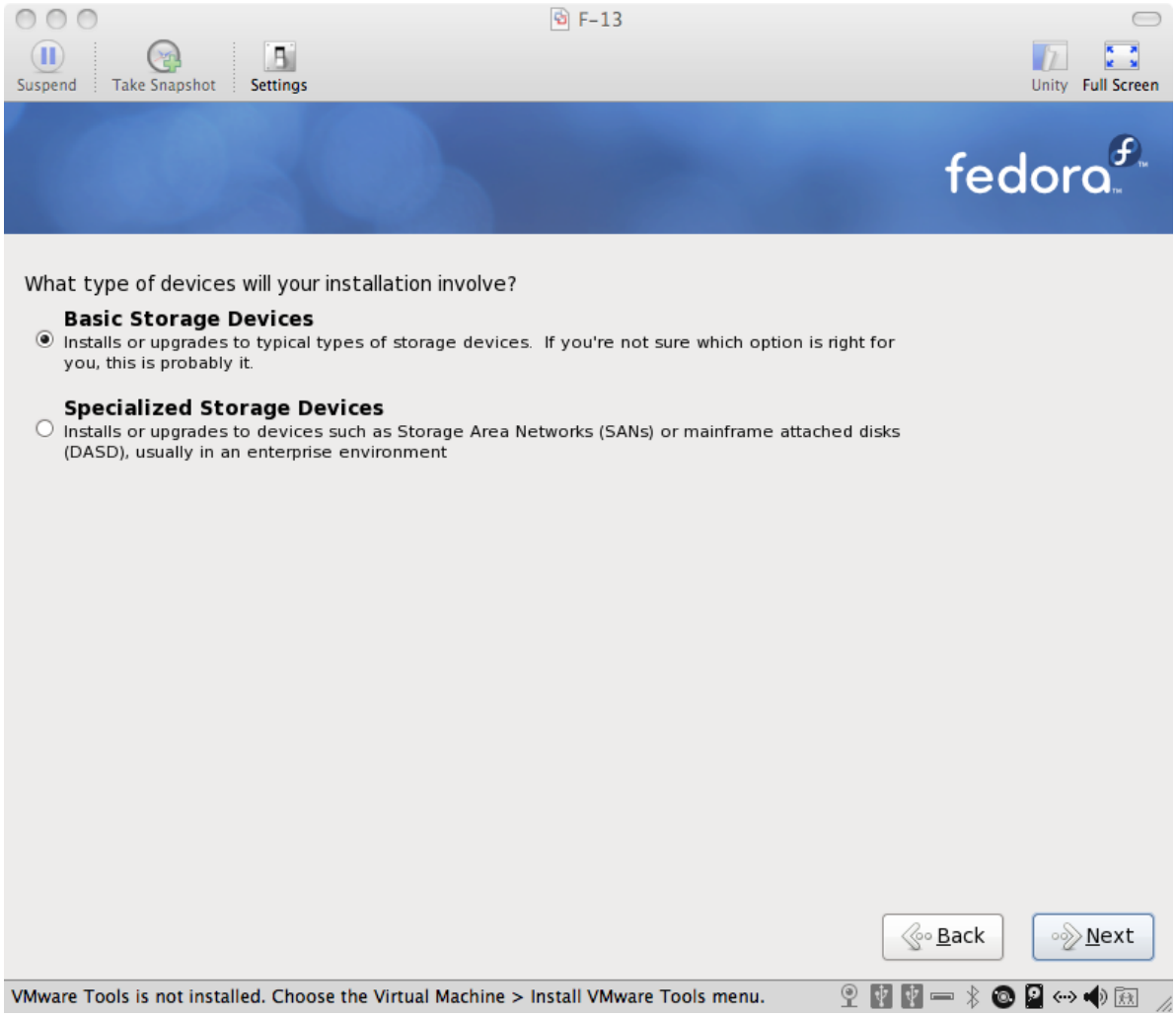
Burn the disk image to a DVD [1] and boot from it. Or use the image to boot a virtual machine as I have done here. After clicking through the welcome screen, select your language and keyboard layout [2]



Fedora Installation: Good choice

Figure 2.1. Fedora Installation - Welcome

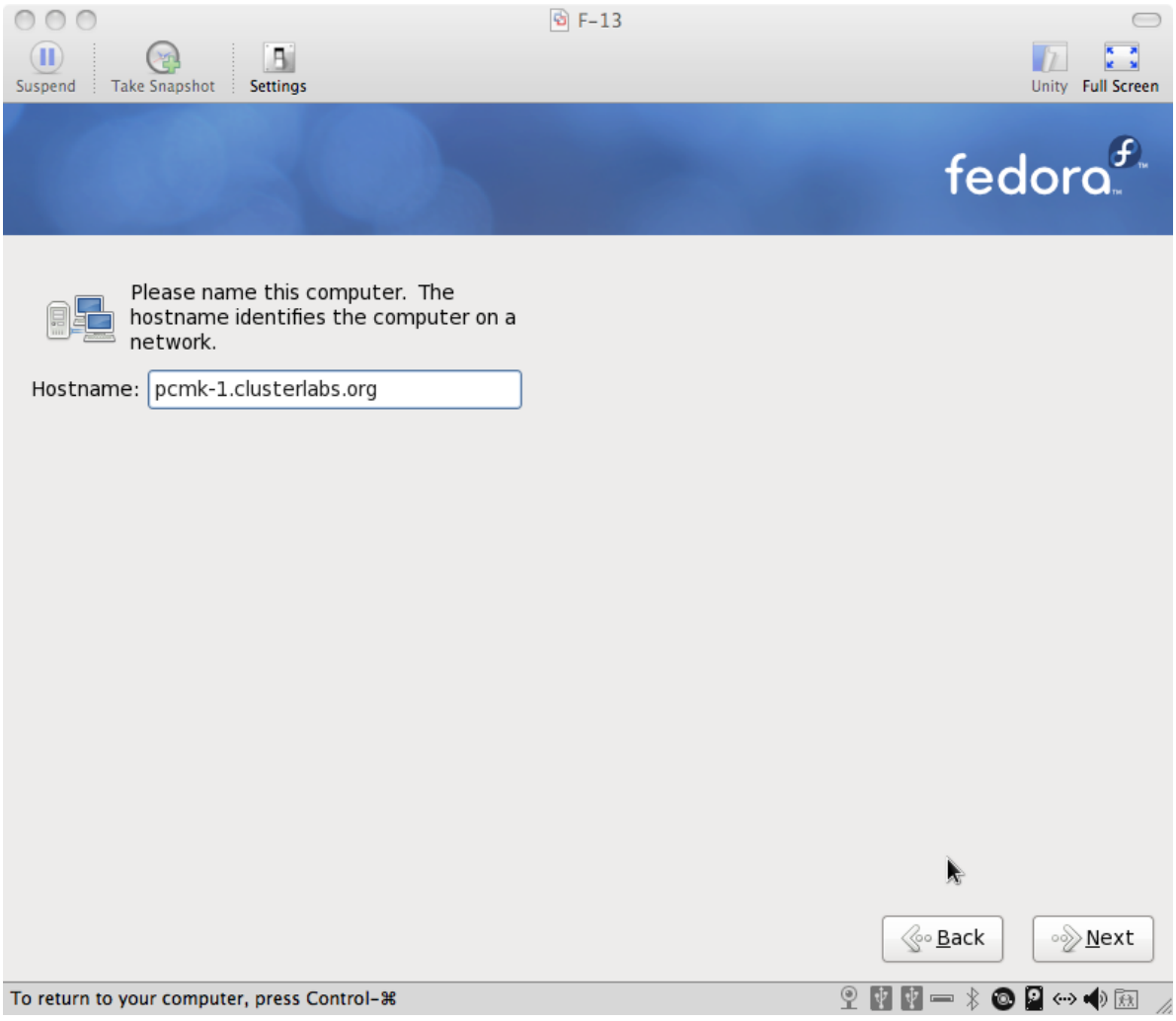
[1] *http://docs.fedoraproject.org/readme-burning-isos/en-US.html* [http://docs.fedoraproject.org/readme-burning-isos/en-US.html]
[2] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/s1-langselection-x86.html* [http://docs.fedoraproject.org/install-guide/f13/en-US/html/s1-langselection-x86.html]



Fedora Installation: Storage Devices

Figure 2.2. Fedora Installation - Storage Devices

Assign your machine a host name. [3] I happen to control the clusterlabs.org domain name, so I will use that here.



Fedora Installation: Choose a hostname

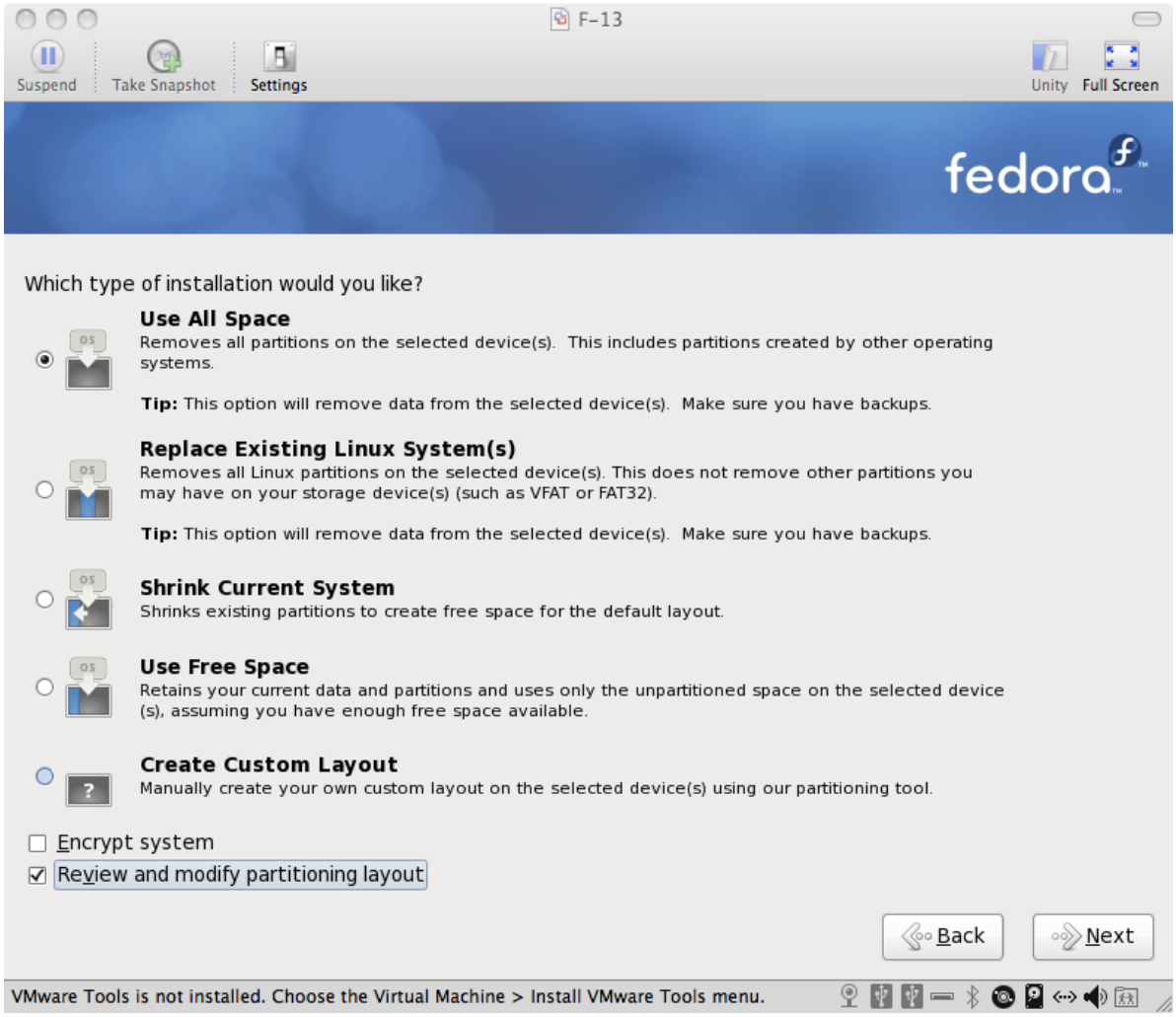
Figure 2.3. Fedora Installation - Hostname

You will then be prompted to indicate the machine's physical location and to supply a root password. [4]

[3] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/sn-networkconfig-fedora.html* [http://docs.fedoraproject.org/install-guide/f13/en-US/html/sn-networkconfig-fedora.html]

[4] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/sn-account_configuration.html* [http://docs.fedoraproject.org/install-guide/f13/en-US/html/sn-account_configuration.html]

Now select where you want Fedora installed. [5] As I don't care about any existing data, I will accept the default and allow Fedora to use the complete drive. However I want to reserve some space for DRBD, so I'll check the **`Review and modify partitioning layout`** box.



Fedora Installation: Choose an installation type

Figure 2.4. Fedora Installation - Installation Type

[5] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/s1-diskpartsetup-x86.html*

By default, Fedora will give all the space to the / (aka. root) partition. Wel'll take some back so we can use DRBD.

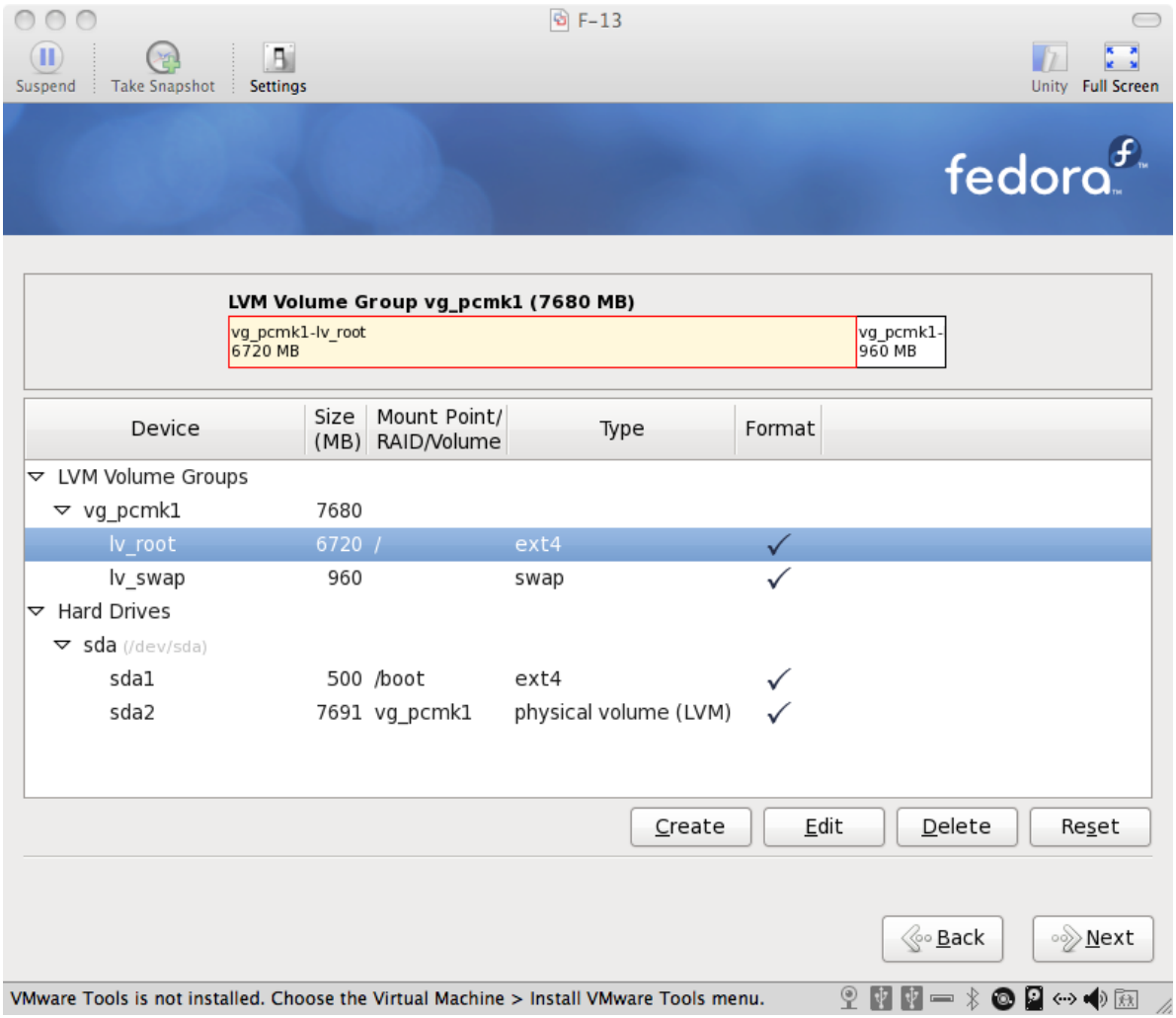
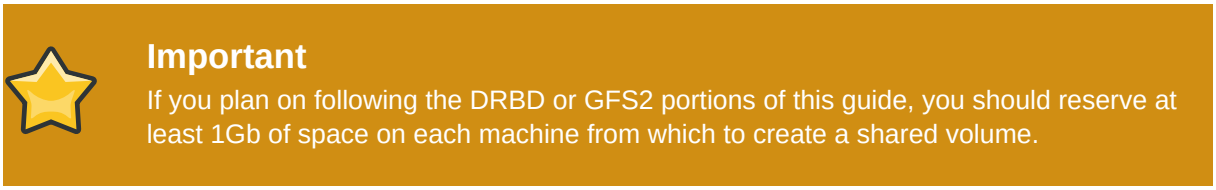


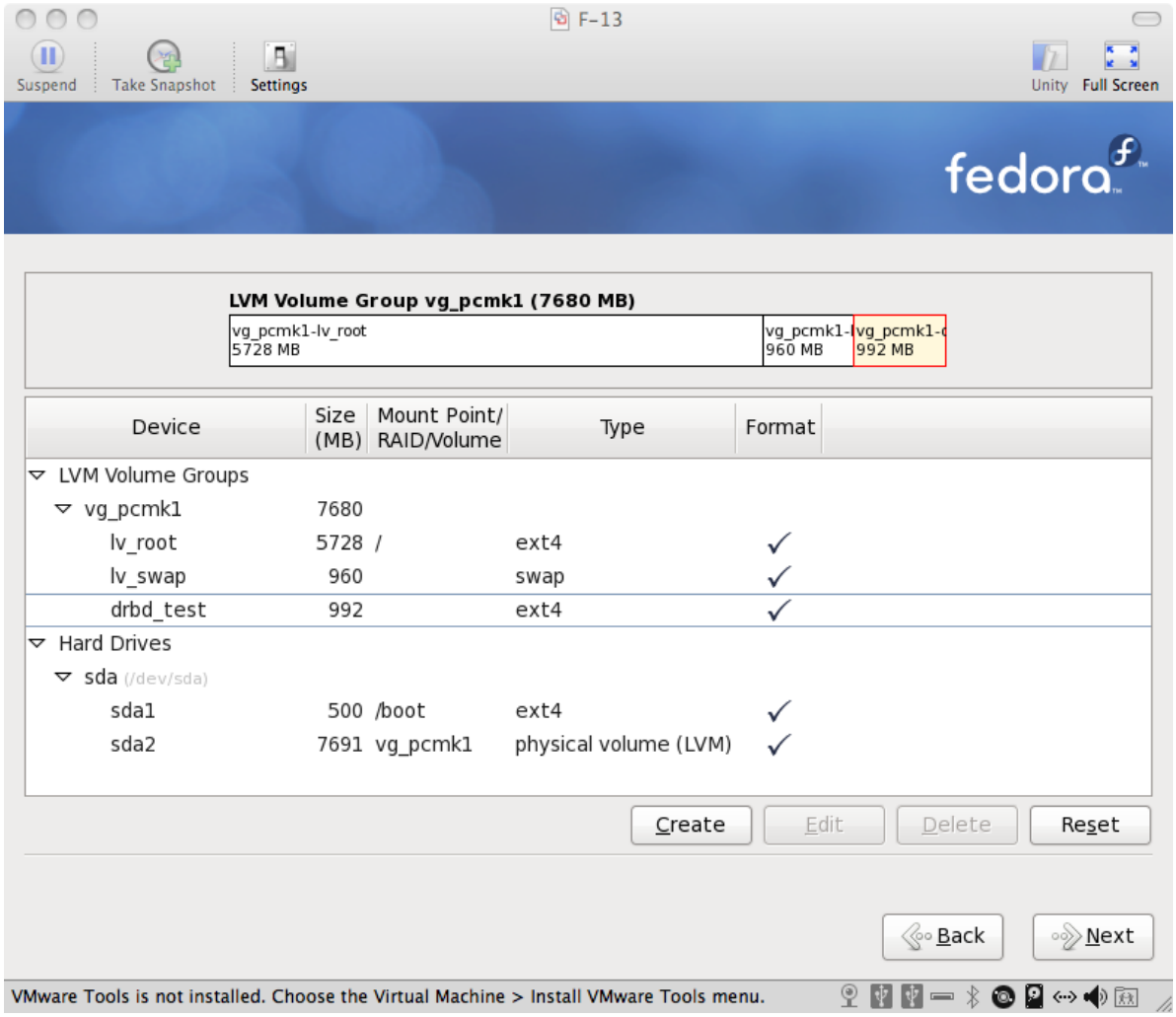
Figure 2.5. Fedora Installation - Default Partitioning

The finalized partition layout should look something like the diagram below.



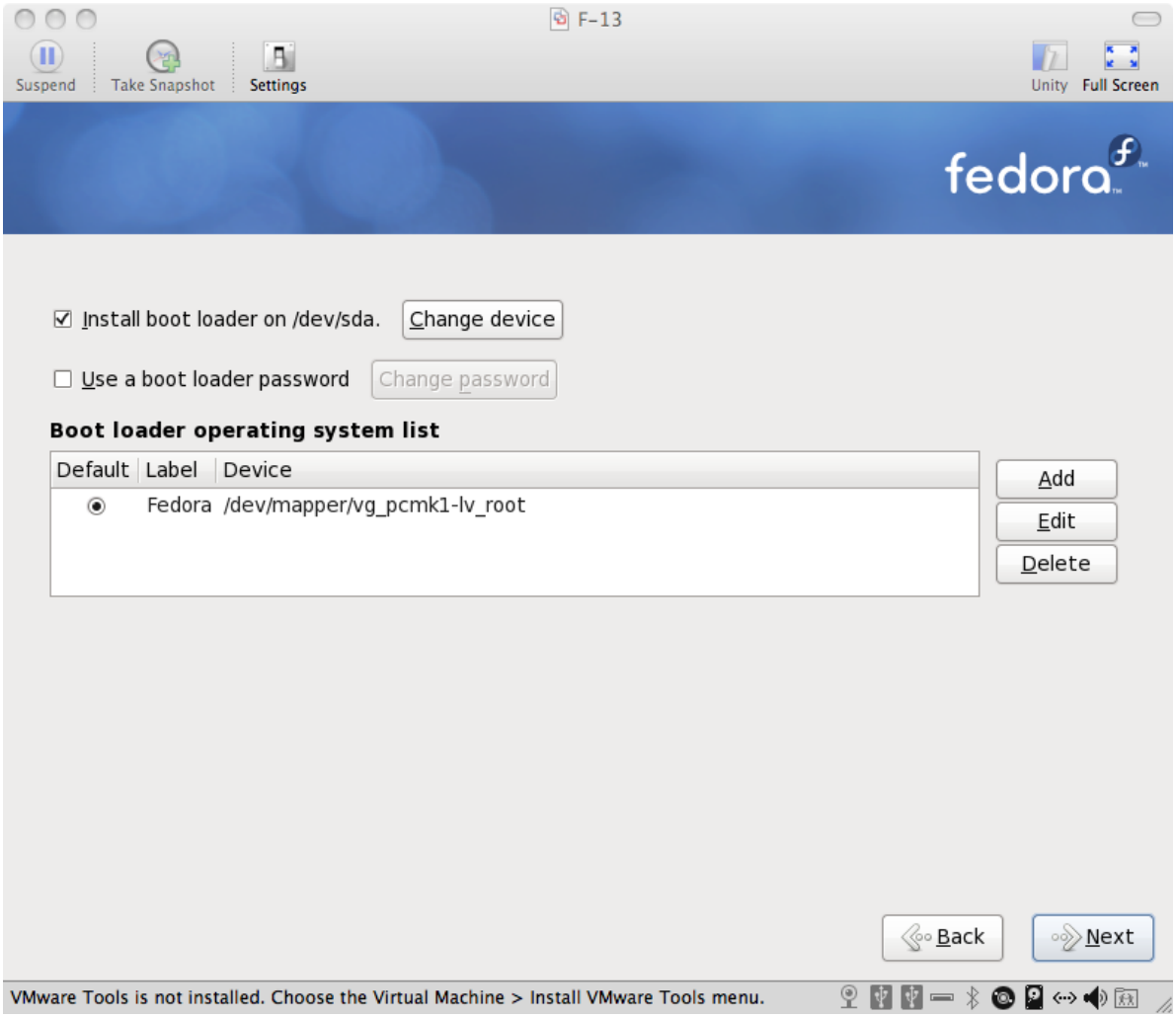
Important

If you plan on following the DRBD or GFS2 portions of this guide, you should reserve at least 1Gb of space on each machine from which to create a shared volume.



Fedora Installation: Create a partition to use (later) for website data

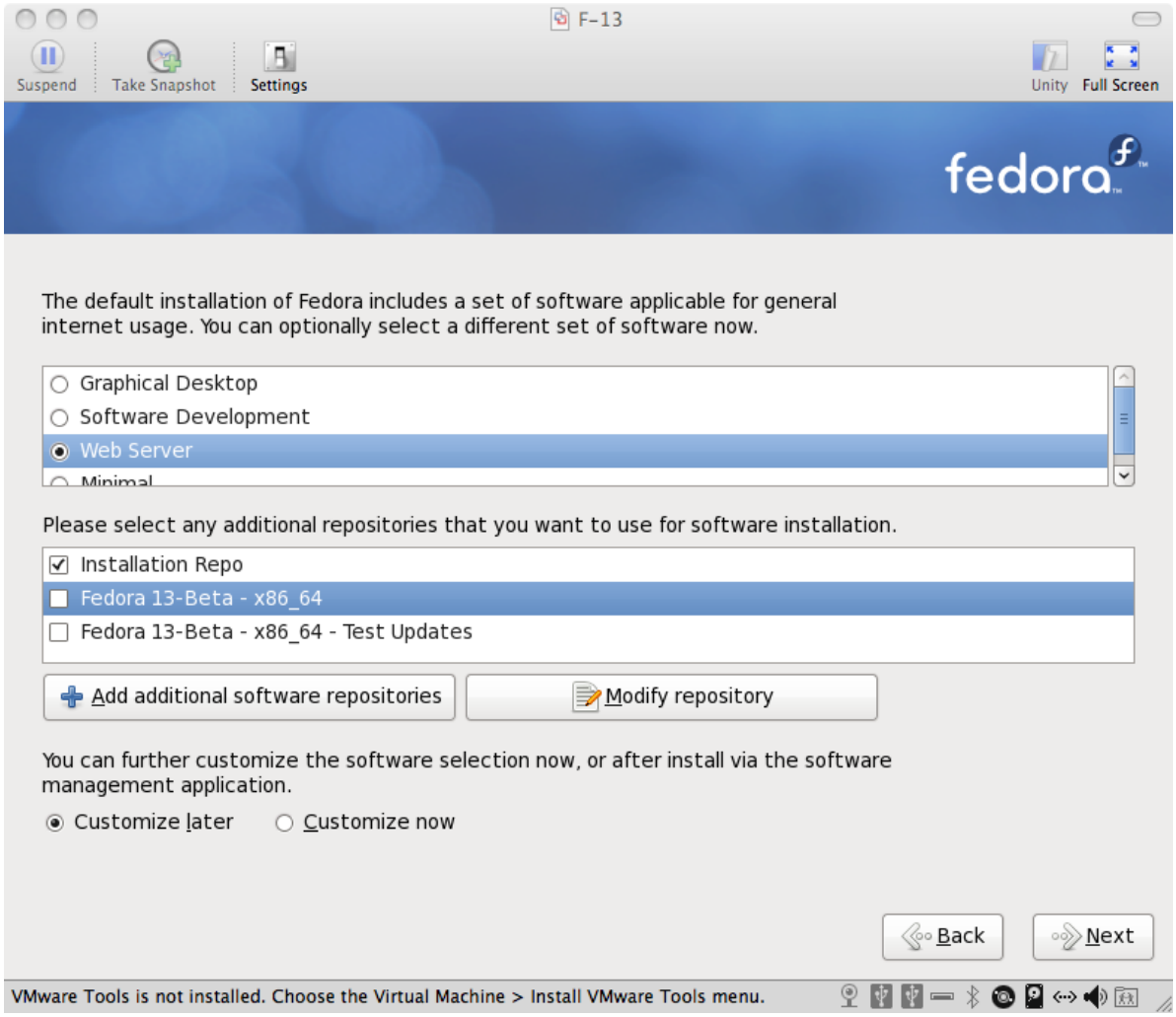
Figure 2.6. Fedora Installation - Customize Partitioning



Fedora Installation: Unless you have a strong reason not to, accept the default bootloader location

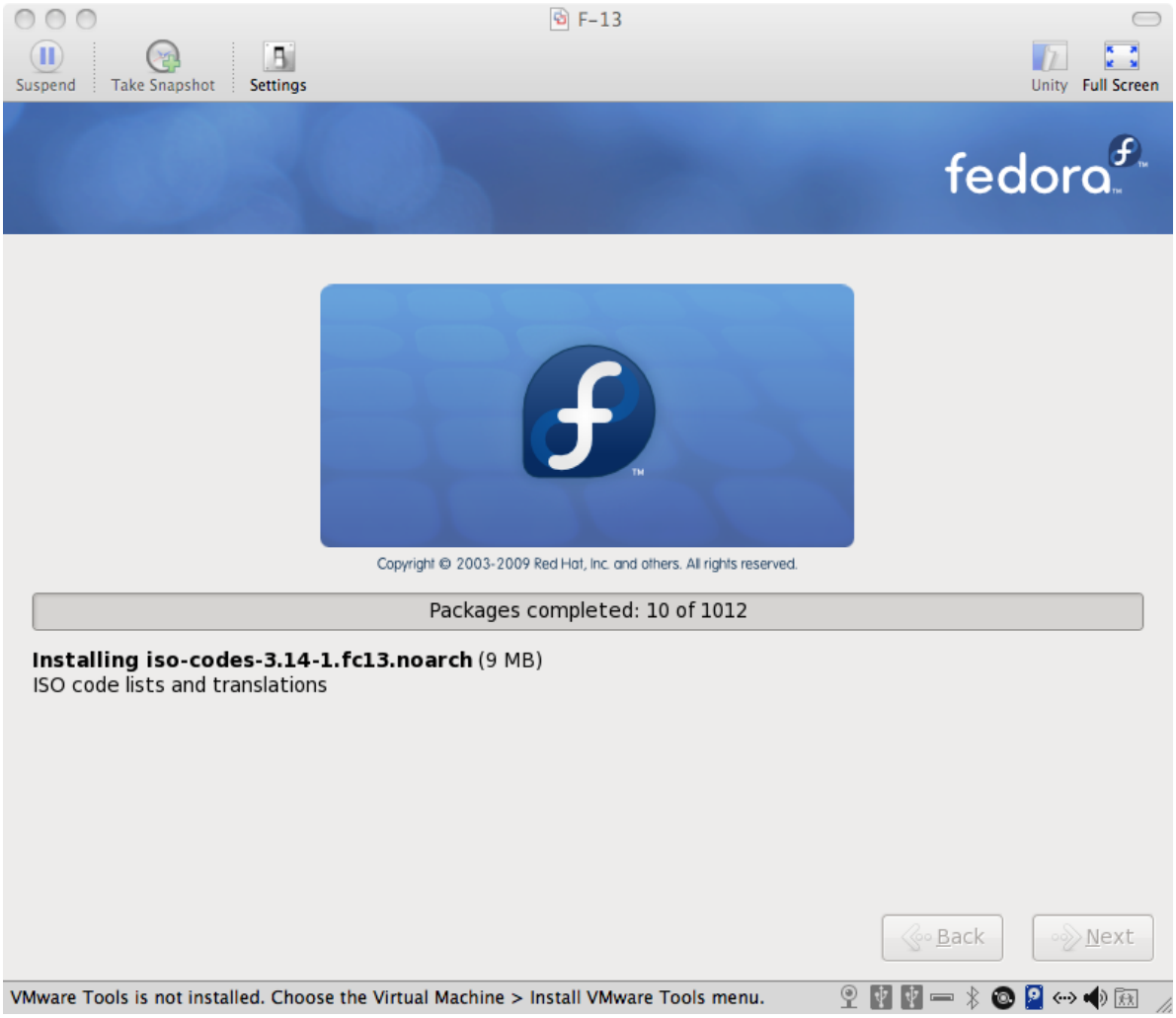
Figure 2.7. Fedora Installation - Bootloader

Next choose which software should be installed. Change the selection to **`Web Server`** since we plan on using Apache. Don't enable updates yet, we'll do that (and install any extra software we need) later. After you click next, Fedora will begin installing.



Fedora Installation: Software selection

Figure 2.8. Fedora Installation - Software



Fedora Installation: Go grab something to drink, this may take a while

Figure 2.9. Fedora Installation - Installing

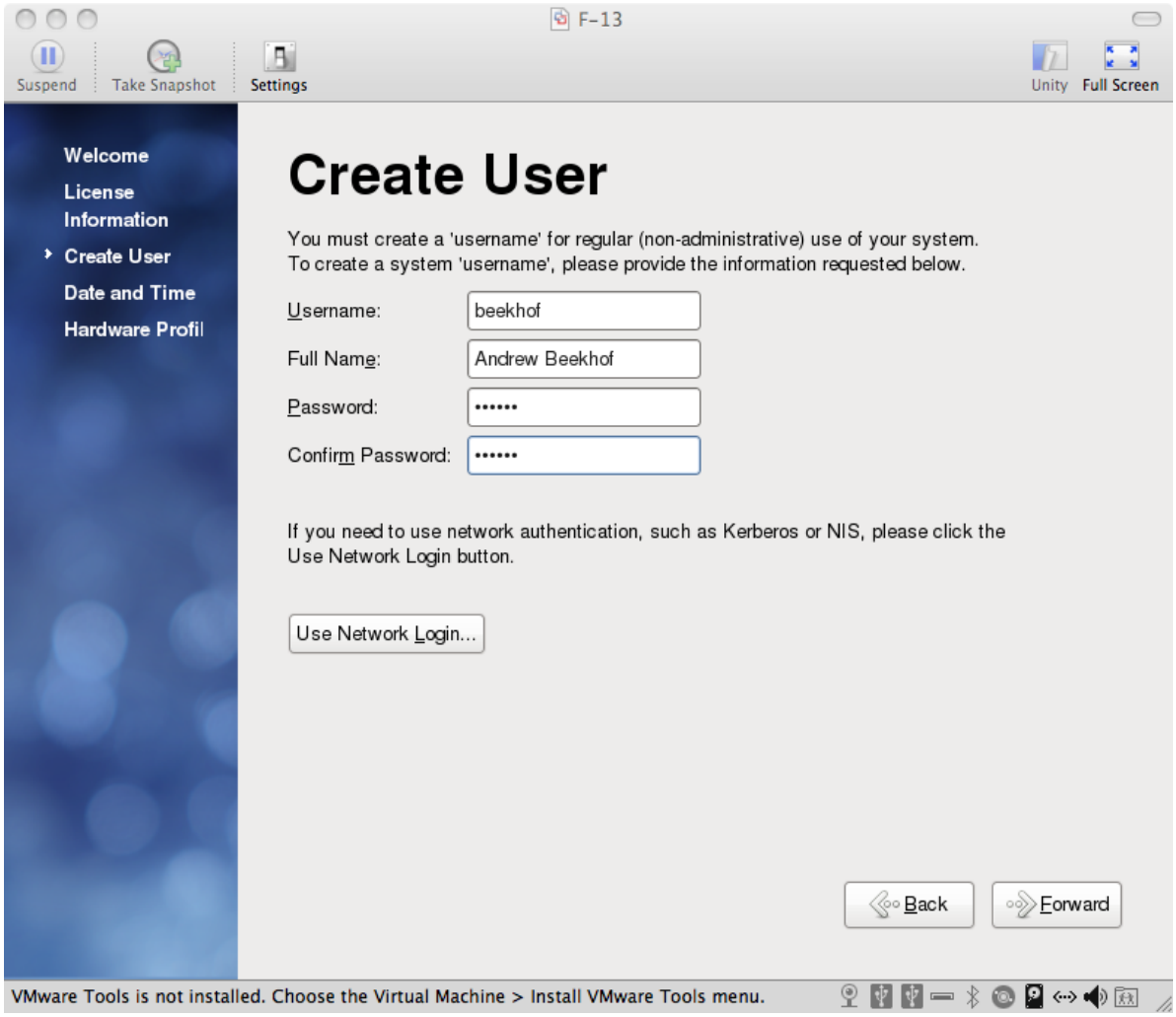
Fedora Installation: Stage 1, completed

Figure 2.10. Fedora Installation - Installation Complete

Once the node reboots, follow the on screen instructions [6] to create a system user and configure the time.

Figure 2.11. Fedora Installation - First Boot

[6] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/ch-firstboot.html* [http://docs.fedoraproject.org/install-guide/f13/en-US/html/ch-firstboot.html]

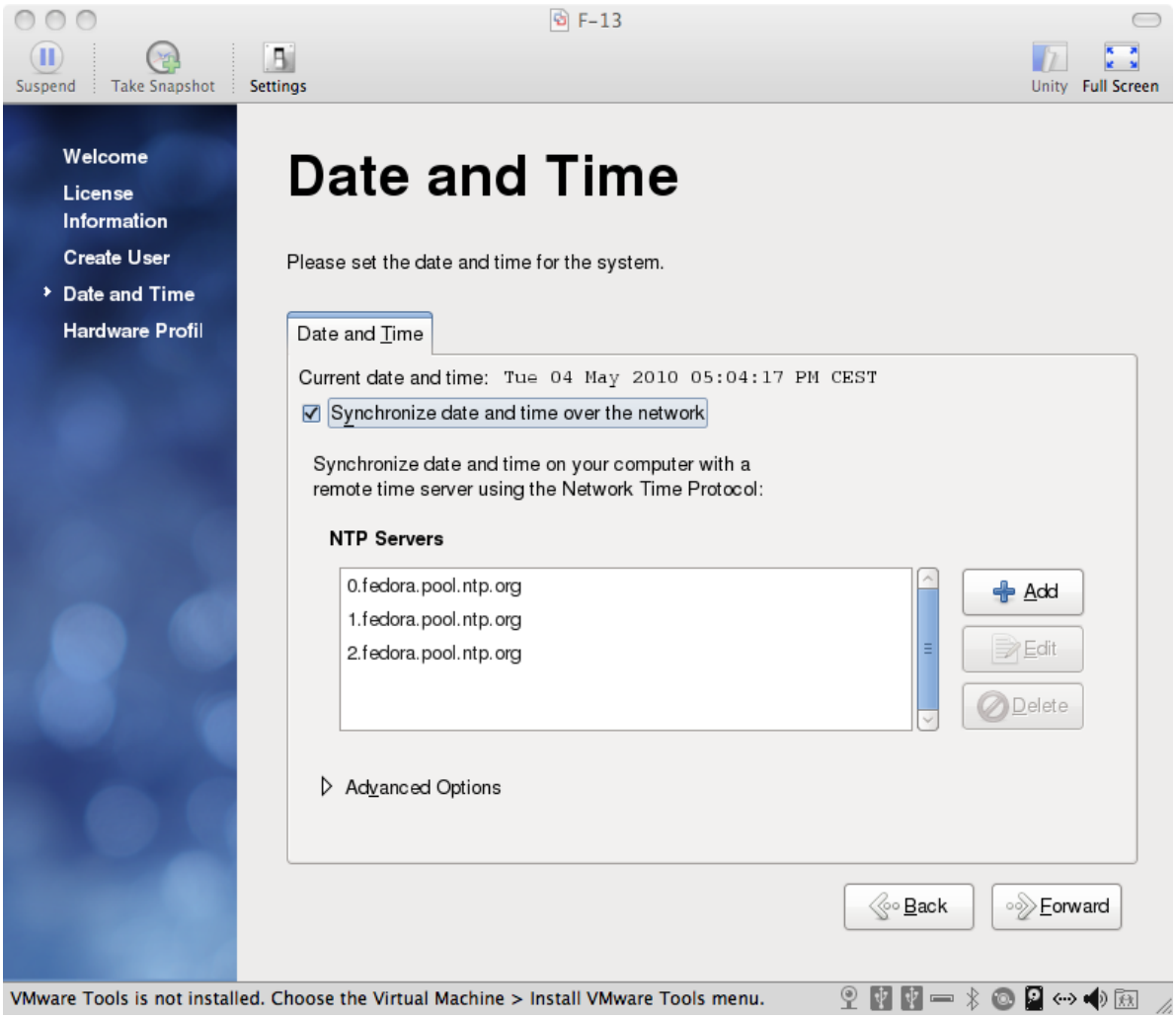


Fedora Installation: Creating a non-privileged user, take note of the password, you'll need it soon

Figure 2.12. Fedora Installation - Create Non-privileged User

Note

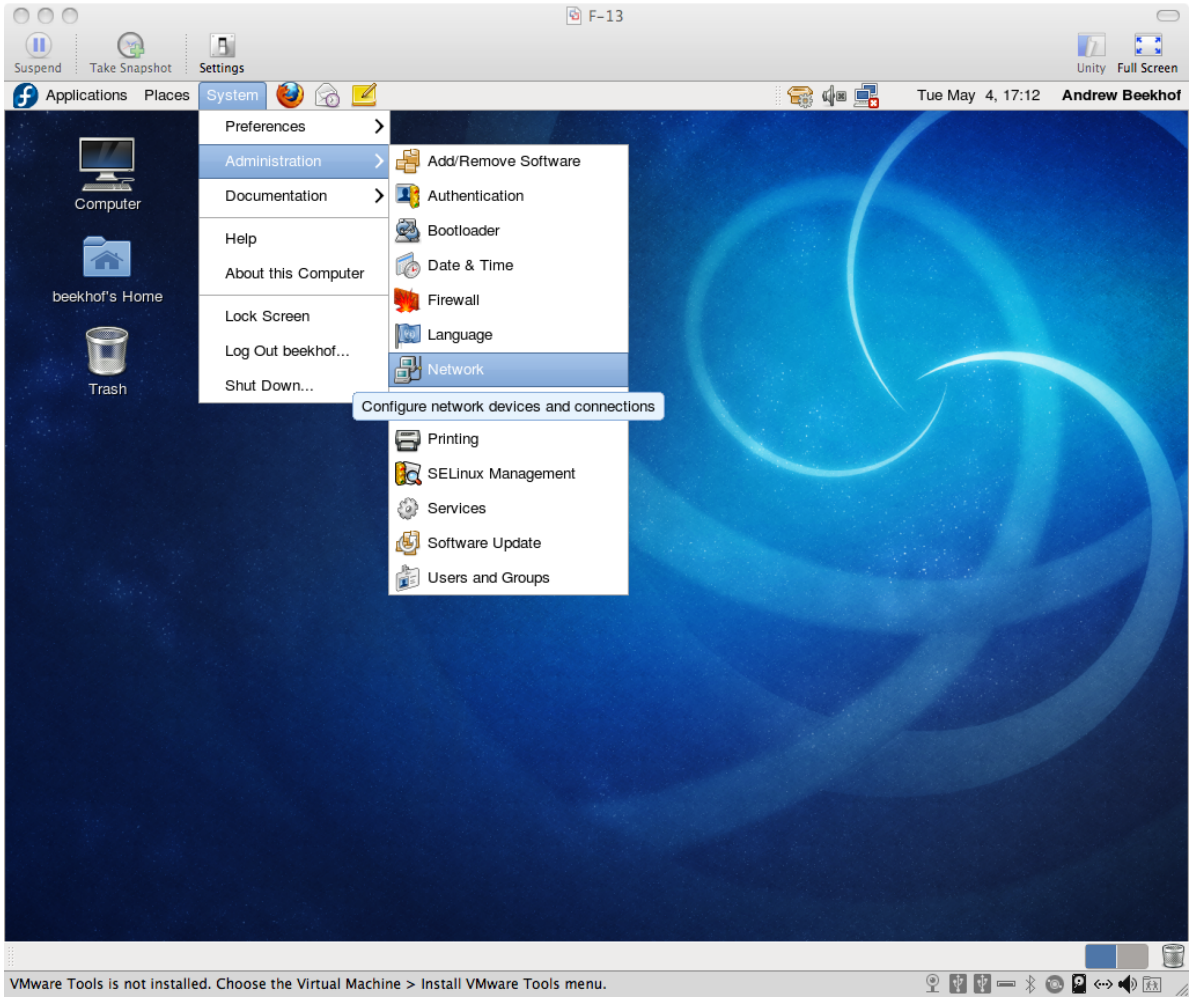
It is highly recommended to enable NTP on your cluster nodes. Doing so ensures all nodes agree on the current time and makes reading log files significantly easier.



Fedora Installation: Enable NTP to keep the times on all your nodes consistent

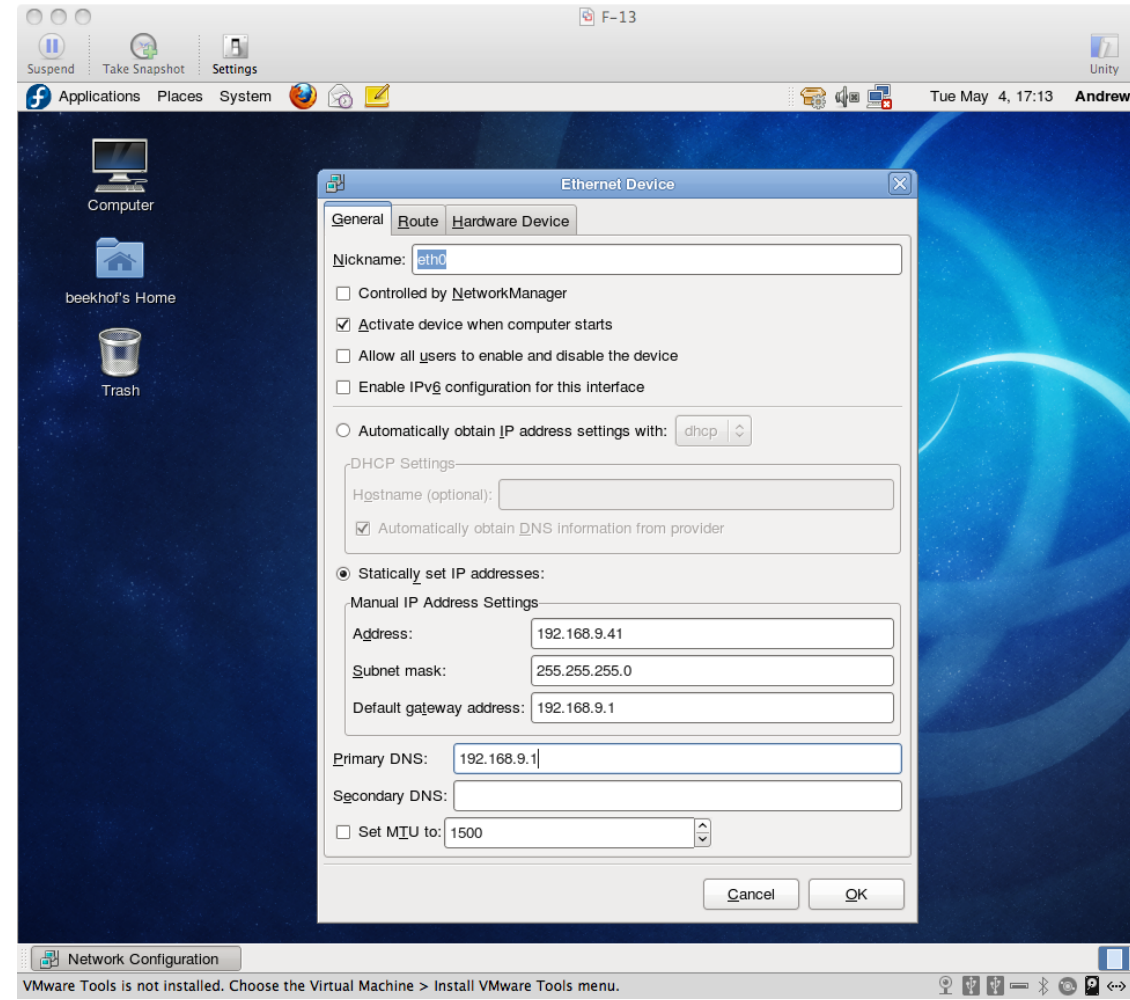
Figure 2.13. Fedora Installation - Date and Time

Click through the next screens until you reach the login window. Click on the user you created and supply the password you indicated earlier.



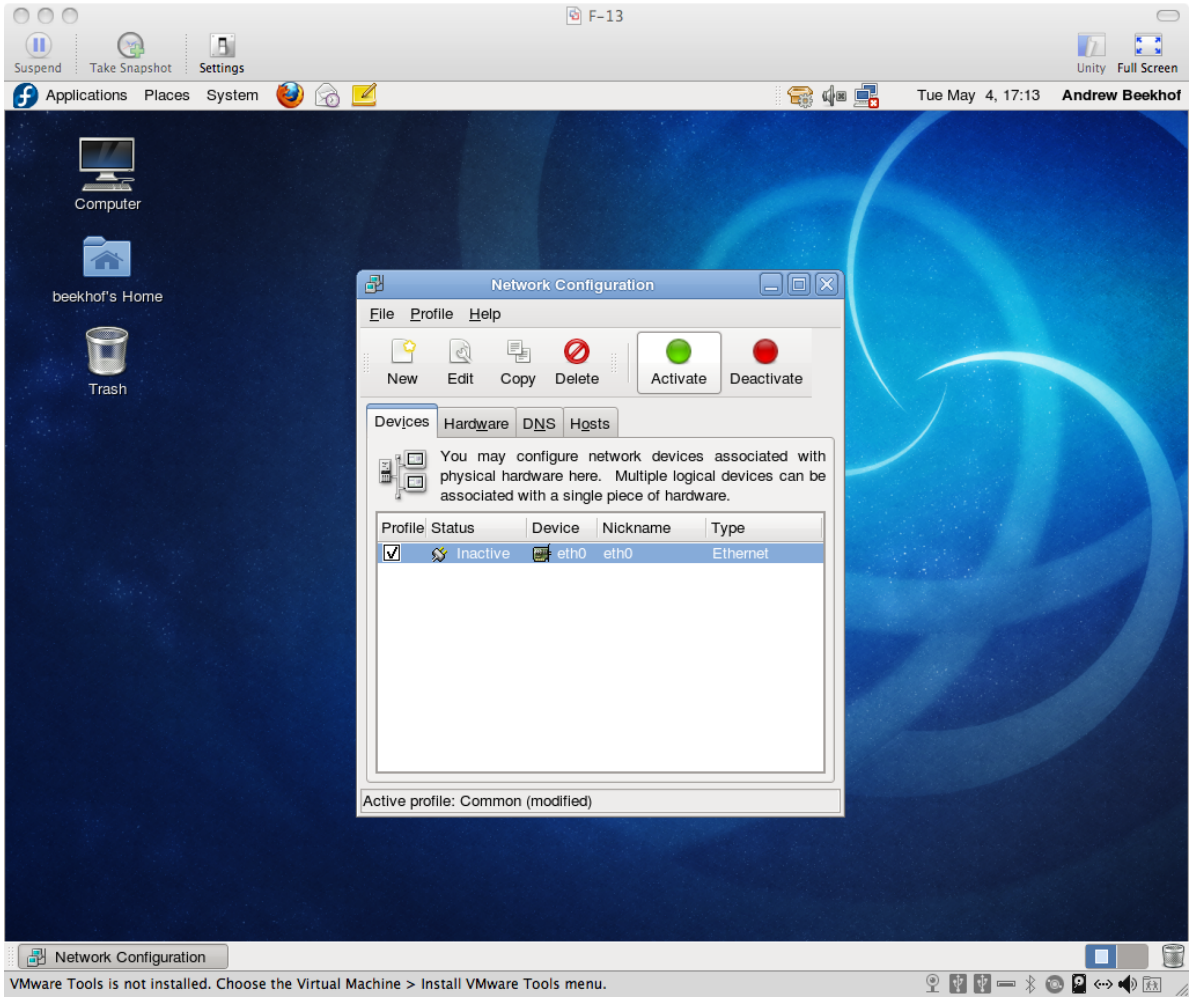
Fedora Installation: Click here to configure networking

Figure 2.14. Fedora Installation - Customize Networking



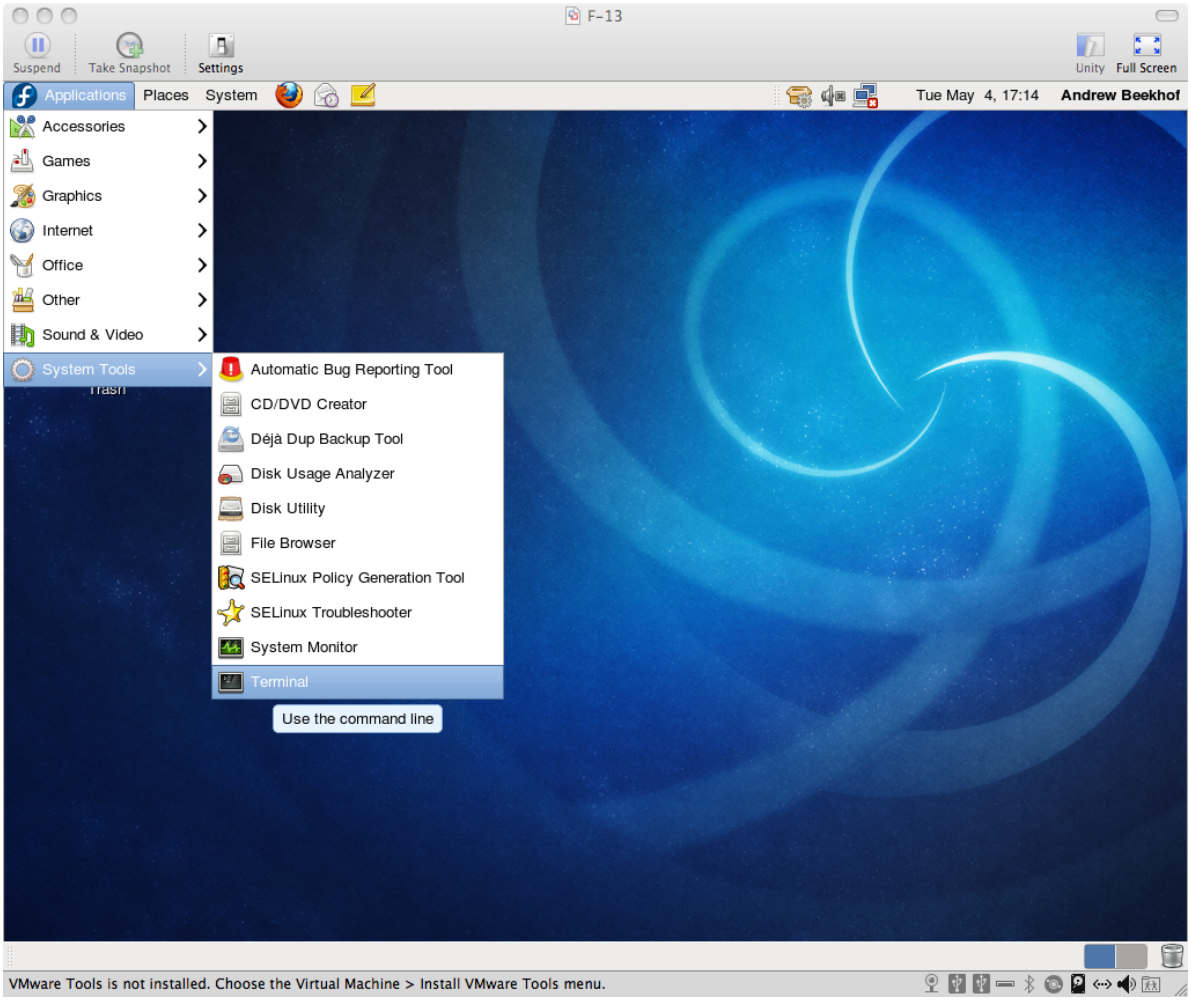
Fedora Installation: Specify network settings for your machine, never choose DHCP

Figure 2.15. Fedora Installation - Specify Network Preferences



Fedora Installation: Click the big green button to activate your changes

Figure 2.16. Fedora Installation - Activate Networking



Fedora Installation: Down to business, fire up the command line

Figure 2.17. Fedora Installation - Bring up the Terminal

Note

That was the last screenshot, from here on in we're going to be working from the terminal.

2.2. Cluster Software Installation

Go to the terminal window you just opened and switch to the super user (aka. "root") account with the **su** command. You will need to supply the password you entered earlier during the installation process.

```
[beekhof@pcmk-1 ~]$ su -
Password:
[root@pcmk-1 ~]#
```

Note

Note that the username (the text before the @ symbol) now indicates we're running as the super user "root".

```
[root@pcmk-1 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:6f:e1:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.9.41/24 brd 192.168.9.255 scope global eth0
    inet6 ::20c:29ff:fe6f:e158/64 scope global dynamic
       valid_lft 2591667sec preferred_lft 604467sec
    inet6 2002:57ae:43fc:0:20c:29ff:fe6f:e158/64 scope global dynamic
       valid_lft 2591990sec preferred_lft 604790sec
    inet6 fe80::20c:29ff:fe6f:e158/64 scope link
       valid_lft forever preferred_lft forever
[root@pcmk-1 ~]# ping -c 1 www.google.com
PING www.l.google.com (74.125.39.99) 56(84) bytes of data.
64 bytes from fx-in-f99.1e100.net (74.125.39.99): icmp_seq=1 ttl=56 time=16.7 ms

--- www.l.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 20ms
rtt min/avg/max/mdev = 16.713/16.713/16.713/0.000 ms
[root@pcmk-1 ~]# /sbin/chkconfig network on
[root@pcmk-1 ~]#
```

2.2.1. Security Shortcuts

To simplify this guide and focus on the aspects directly connected to clustering, we will now disable the machine's firewall and SELinux installation. Both of these actions create significant security issues and should not be performed on machines that will be exposed to the outside world.

Important

TODO: Create an Appendix that deals with (at least) re-enabling the firewall.

```
[root@pcmk-1 ~]# sed -i.bak "s/SELINUX=enforcing/SELINUX=permissive/g" /etc/selinux/config
[root@pcmk-1 ~]# /sbin/chkconfig --del iptables
[root@pcmk-1 ~]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
```

Important

You will need to reboot for the SELinux changes to take effect. Otherwise you will see something like this when you start corosync:

```
May  4 19:30:54 pcmk-1 setroubleshoot: SELinux is preventing /usr/sbin/
corosync "getattr" access on /. For complete SELinux messages. run sealert -l
 6e0d4384-638e-4d55-9aaf-7dac011f29c1
May  4 19:30:54 pcmk-1 setroubleshoot: SELinux is preventing /usr/sbin/
corosync "getattr" access on /. For complete SELinux messages. run sealert -l
 6e0d4384-638e-4d55-9aaf-7dac011f29c1
```

2.2.2. Install the Cluster Software

Since version 12, Fedora comes with recent versions of everything you need, so simply fire up the shell and run:

```
[root@pcmk-1 ~]# sed -i.bak "s/enabled=0/enabled=1/g" /etc/yum.repos.d/fedora.repo
[root@pcmk-1 ~]# sed -i.bak "s/enabled=0/enabled=1/g" /etc/yum.repos.d/fedora-updates.repo
[root@pcmk-1 ~]# yum install -y pacemaker corosync
Loaded plugins: presto, refresh-packagekit
fedora/metalink                                          |  22 kB     00:00
fedora-debuginfo/metalink                                |  16 kB     00:00
fedora-debuginfo                                         | 3.2 kB     00:00
fedora-debuginfo/primary_db                              | 1.4 MB     00:04
fedora-source/metalink                                   |  22 kB     00:00
fedora-source                                            | 3.2 kB     00:00
fedora-source/primary_db                                 | 3.0 MB     00:05
updates/metalink                                         |  26 kB     00:00
updates                                                  | 2.6 kB     00:00
updates/primary_db                                       | 1.1 kB     00:00
updates-debuginfo/metalink                               |  18 kB     00:00
updates-debuginfo                                        | 2.6 kB     00:00
updates-debuginfo/primary_db                             | 1.1 kB     00:00
updates-source/metalink                                  |  25 kB     00:00
updates-source                                           | 2.6 kB     00:00
updates-source/primary_db                                | 1.1 kB     00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package corosync.x86_64 0:1.2.1-1.fc13 set to be updated
--> Processing Dependency: corosynclib = 1.2.1-1.fc13 for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libquorum.so.4(COROSYNC_QUORUM_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libvotequorum.so.4(COROSYNC_VOTEQUORUM_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcpg.so.4(COROSYNC_CPG_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libconfdb.so.4(COROSYNC_CONFDB_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcfg.so.4(COROSYNC_CFG_0.82)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libpload.so.4(COROSYNC_PLOAD_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: liblogsys.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libconfdb.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcoroipcc.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcpg.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libquorum.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcoroipcs.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
```

```
--> Processing Dependency: libvotequorum.so.4()(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcfg.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libtotem_pg.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libpload.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
---> Package pacemaker.x86_64 0:1.1.1-1.fc13 set to be updated
--> Processing Dependency: heartbeat >= 3.0.0 for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: net-snmp >= 5.4 for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: resource-agents for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: cluster-glue for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libnetsnmp.so.20()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libcrmcluster.so.1()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libpengine.so.3()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libnetsnmpagent.so.20()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libesmtp.so.5()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libstonithd.so.1()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libhbclient.so.1()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libpils.so.2()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libpe_status.so.2()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libnetsnmpmibs.so.20()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libnetsnmphelpers.so.20()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libcib.so.1()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libccmclient.so.1()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libstonith.so.1()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: liblrm.so.2()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libtransitioner.so.1()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libpe_rules.so.2()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libcrmcommon.so.2()(64bit) for package:
 pacemaker-1.1.1-1.fc13.x86_64
--> Processing Dependency: libplumb.so.2()(64bit) for package: pacemaker-1.1.1-1.fc13.x86_64
--> Running transaction check
---> Package cluster-glue.x86_64 0:1.0.2-1.fc13 set to be updated
--> Processing Dependency: perl-TimeDate for package: cluster-glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libOpenIPMIutils.so.0()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libOpenIPMIposix.so.0()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libopenhpi.so.2()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libOpenIPMI.so.0()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
---> Package cluster-glue-libs.x86_64 0:1.0.2-1.fc13 set to be updated
---> Package corosynclib.x86_64 0:1.2.1-1.fc13 set to be updated
--> Processing Dependency: librdmacm.so.1(RDMACM_1.0)(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: libibverbs.so.1(IBVERBS_1.0)(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: libibverbs.so.1(IBVERBS_1.1)(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: libibverbs.so.1()(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: librdmacm.so.1()(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
```

```
---> Package heartbeat.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13 set to be updated
--> Processing Dependency: PyXML for package: heartbeat-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64
---> Package heartbeat-libs.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13 set to be updated
---> Package libesmtp.x86_64 0:1.0.4-12.fc12 set to be updated
---> Package net-snmp.x86_64 1:5.5-12.fc13 set to be updated
--> Processing Dependency: libsensors.so.4()(64bit) for package: 1:net-snmp-5.5-12.fc13.x86_64
---> Package net-snmp-libs.x86_64 1:5.5-12.fc13 set to be updated
---> Package pacemaker-libs.x86_64 0:1.1.1-1.fc13 set to be updated
---> Package resource-agents.x86_64 0:3.0.10-1.fc13 set to be updated
--> Processing Dependency: libnet.so.1()(64bit) for package: resource-
agents-3.0.10-1.fc13.x86_64
--> Running transaction check
---> Package OpenIPMI-libs.x86_64 0:2.0.16-8.fc13 set to be updated
---> Package PyXML.x86_64 0:0.8.4-17.fc13 set to be updated
---> Package libibverbs.x86_64 0:1.1.3-4.fc13 set to be updated
--> Processing Dependency: libibverbs-driver for package: libibverbs-1.1.3-4.fc13.x86_64
---> Package libnet.x86_64 0:1.1.4-3.fc12 set to be updated
---> Package librdmacm.x86_64 0:1.0.10-2.fc13 set to be updated
---> Package lm_sensors-libs.x86_64 0:3.1.2-2.fc13 set to be updated
---> Package openhpi-libs.x86_64 0:2.14.1-3.fc13 set to be updated
---> Package perl-TimeDate.noarch 1:1.20-1.fc13 set to be updated
--> Running transaction check
---> Package libmlx4.x86_64 0:1.0.1-5.fc13 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

```
================================================================================
 Package                Arch     Version                            Repository      Size
================================================================================
Installing:
 corosync               x86_64   1.2.1-1.fc13                       fedora         136 k
 pacemaker              x86_64   1.1.1-1.fc13                       fedora         543 k
Installing for dependencies:
 OpenIPMI-libs          x86_64   2.0.16-8.fc13                      fedora         474 k
 PyXML                  x86_64   0.8.4-17.fc13                      fedora         906 k
 cluster-glue           x86_64   1.0.2-1.fc13                       fedora         230 k
 cluster-glue-libs      x86_64   1.0.2-1.fc13                       fedora         116 k
 corosynclib            x86_64   1.2.1-1.fc13                       fedora         145 k
 heartbeat              x86_64   3.0.0-0.7.0daab7da36a8.hg.fc13     updates        172 k
 heartbeat-libs         x86_64   3.0.0-0.7.0daab7da36a8.hg.fc13     updates        265 k
 libesmtp               x86_64   1.0.4-12.fc12                      fedora          54 k
 libibverbs             x86_64   1.1.3-4.fc13                       fedora          42 k
 libmlx4                x86_64   1.0.1-5.fc13                       fedora          27 k
 libnet                 x86_64   1.1.4-3.fc12                       fedora          49 k
 librdmacm              x86_64   1.0.10-2.fc13                      fedora          22 k
 lm_sensors-libs        x86_64   3.1.2-2.fc13                       fedora          37 k
 net-snmp               x86_64   1:5.5-12.fc13                      fedora         295 k
 net-snmp-libs          x86_64   1:5.5-12.fc13                      fedora         1.5 M
 openhpi-libs           x86_64   2.14.1-3.fc13                      fedora         135 k
 pacemaker-libs         x86_64   1.1.1-1.fc13                       fedora         264 k
 perl-TimeDate          noarch   1:1.20-1.fc13                      fedora          42 k
 resource-agents        x86_64   3.0.10-1.fc13                      fedora         357 k

Transaction Summary
================================================================================
Install      21 Package(s)
Upgrade       0 Package(s)

Total download size: 5.7 M
Installed size: 20 M
Downloading Packages:
```

```
Setting up and reading Presto delta metadata
updates-testing/prestodelta                                | 164 kB     00:00
fedora/prestodelta                                         |  150 B     00:00
Processing delta metadata
Package(s) data still to download: 5.7 M
(1/21): OpenIPMI-libs-2.0.16-8.fc13.x86_64.rpm             | 474 kB     00:00
(2/21): PyXML-0.8.4-17.fc13.x86_64.rpm                     | 906 kB     00:01
(3/21): cluster-glue-1.0.2-1.fc13.x86_64.rpm               | 230 kB     00:00
(4/21): cluster-glue-libs-1.0.2-1.fc13.x86_64.rpm          | 116 kB     00:00
(5/21): corosync-1.2.1-1.fc13.x86_64.rpm                   | 136 kB     00:00
(6/21): corosynclib-1.2.1-1.fc13.x86_64.rpm                | 145 kB     00:00
(7/21): heartbeat-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64.rpm | 172 kB     00:00
(8/21): heartbeat-libs-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64.rpm | 265 kB     00:00
(9/21): libesmtp-1.0.4-12.fc12.x86_64.rpm                  |  54 kB     00:00
(10/21): libibverbs-1.1.3-4.fc13.x86_64.rpm                |  42 kB     00:00
(11/21): libmlx4-1.0.1-5.fc13.x86_64.rpm                   |  27 kB     00:00
(12/21): libnet-1.1.4-3.fc12.x86_64.rpm                    |  49 kB     00:00
(13/21): librdmacm-1.0.10-2.fc13.x86_64.rpm                |  22 kB     00:00
(14/21): lm_sensors-libs-3.1.2-2.fc13.x86_64.rpm           |  37 kB     00:00
(15/21): net-snmp-5.5-12.fc13.x86_64.rpm                   | 295 kB     00:00
(16/21): net-snmp-libs-5.5-12.fc13.x86_64.rpm              | 1.5 MB     00:01
(17/21): openhpi-libs-2.14.1-3.fc13.x86_64.rpm             | 135 kB     00:00
(18/21): pacemaker-1.1.1-1.fc13.x86_64.rpm                 | 543 kB     00:00
(19/21): pacemaker-libs-1.1.1-1.fc13.x86_64.rpm            | 264 kB     00:00
(20/21): perl-TimeDate-1.20-1.fc13.noarch.rpm              |  42 kB     00:00
(21/21): resource-agents-3.0.10-1.fc13.x86_64.rpm          | 357 kB     00:00
--------------------------------------------------------------------------------
Total                                              539 kB/s | 5.7 MB     00:10
warning: rpmts_HdrFromFdno: Header V3 RSA/SHA256 Signature, key ID e8e40fde: NOKEY
fedora/gpgkey                                              | 3.2 kB     00:00 ...
Importing GPG key 0xE8E40FDE "Fedora (13) <fedora@fedoraproject.org%gt;" from /etc/pki/rpm-
gpg/RPM-GPG-KEY-fedora-x86_64
```

```
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : lm_sensors-libs-3.1.2-2.fc13.x86_64                          1/21
  Installing     : 1:net-snmp-libs-5.5-12.fc13.x86_64                           2/21
  Installing     : 1:net-snmp-5.5-12.fc13.x86_64                                3/21
  Installing     : openhpi-libs-2.14.1-3.fc13.x86_64                            4/21
  Installing     : libibverbs-1.1.3-4.fc13.x86_64                               5/21
  Installing     : libmlx4-1.0.1-5.fc13.x86_64                                  6/21
  Installing     : librdmacm-1.0.10-2.fc13.x86_64                               7/21
  Installing     : corosync-1.2.1-1.fc13.x86_64                                 8/21
  Installing     : corosynclib-1.2.1-1.fc13.x86_64                              9/21
  Installing     : libesmtp-1.0.4-12.fc12.x86_64                               10/21
  Installing     : OpenIPMI-libs-2.0.16-8.fc13.x86_64                          11/21
  Installing     : PyXML-0.8.4-17.fc13.x86_64                                  12/21
  Installing     : libnet-1.1.4-3.fc12.x86_64                                  13/21
  Installing     : 1:perl-TimeDate-1.20-1.fc13.noarch                          14/21
  Installing     : cluster-glue-1.0.2-1.fc13.x86_64                            15/21
  Installing     : cluster-glue-libs-1.0.2-1.fc13.x86_64                       16/21
  Installing     : resource-agents-3.0.10-1.fc13.x86_64                        17/21
  Installing     : heartbeat-libs-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64        18/21
  Installing     : heartbeat-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64             19/21
  Installing     : pacemaker-1.1.1-1.fc13.x86_64                               20/21
  Installing     : pacemaker-libs-1.1.1-1.fc13.x86_64                          21/21

Installed:
  corosync.x86_64 0:1.2.1-1.fc13                   pacemaker.x86_64 0:1.1.1-1.fc13
```

```
Dependency Installed:
  OpenIPMI-libs.x86_64 0:2.0.16-8.fc13
  PyXML.x86_64 0:0.8.4-17.fc13
  cluster-glue.x86_64 0:1.0.2-1.fc13
  cluster-glue-libs.x86_64 0:1.0.2-1.fc13
  corosynclib.x86_64 0:1.2.1-1.fc13
  heartbeat.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13
  heartbeat-libs.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13
  libesmtp.x86_64 0:1.0.4-12.fc12
  libibverbs.x86_64 0:1.1.3-4.fc13
  libmlx4.x86_64 0:1.0.1-5.fc13
  libnet.x86_64 0:1.1.4-3.fc12
  librdmacm.x86_64 0:1.0.10-2.fc13
  lm_sensors-libs.x86_64 0:3.1.2-2.fc13
  net-snmp.x86_64 1:5.5-12.fc13
  net-snmp-libs.x86_64 1:5.5-12.fc13
  openhpi-libs.x86_64 0:2.14.1-3.fc13
  pacemaker-libs.x86_64 0:1.1.1-1.fc13
  perl-TimeDate.noarch 1:1.20-1.fc13
  resource-agents.x86_64 0:3.0.10-1.fc13

Complete!
[root@pcmk-1 ~]#
```

2.3. Before You Continue

Repeat the Installation steps so that you have 2 Fedora nodes with the cluster software installed.

For the purposes of this document, the additional node is called pcmk-2 with address 192.168.122.42.

2.4. Setup

2.4.1. Finalize Networking

Confirm that you can communicate with the two new nodes:

```
   ping -c 3 192.168.122.102
[root@pcmk-1 ~]# ping -c 3 192.168.122.102
PING 192.168.122.102 (192.168.122.102) 56(84) bytes of data.
64 bytes from 192.168.122.102: icmp_seq=1 ttl=64 time=0.343 ms
64 bytes from 192.168.122.102: icmp_seq=2 ttl=64 time=0.402 ms
64 bytes from 192.168.122.102: icmp_seq=3 ttl=64 time=0.558 ms

--- 192.168.122.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.343/0.434/0.558/0.092 ms
```

Figure 2.18. Verify Connectivity by IP address

Now we need to make sure we can communicate with the machines by their name. If you have a DNS server, add additional entries for the three machines. Otherwise, you'll need to add the machines to /etc/hosts . Below are the entries for my cluster nodes:

```
   grep pcmk /etc/hosts
[root@pcmk-1 ~]# grep pcmk /etc/hosts
192.168.122.101 pcmk-1.clusterlabs.org pcmk-1
192.168.122.102 pcmk-2.clusterlabs.org pcmk-2
```

Figure 2.19. Set up /etc/hosts entries

We can now verify the setup by again using ping:

```
   ping -c 3 pcmk-2
[root@pcmk-1 ~]# ping -c 3 pcmk-2
PING pcmk-2.clusterlabs.org (192.168.122.101) 56(84) bytes of data.
64 bytes from pcmk-1.clusterlabs.org (192.168.122.101): icmp_seq=1 ttl=64 time=0.164 ms
64 bytes from pcmk-1.clusterlabs.org (192.168.122.101): icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from pcmk-1.clusterlabs.org (192.168.122.101): icmp_seq=3 ttl=64 time=0.186 ms

--- pcmk-2.clusterlabs.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.164/0.275/0.475/0.141 ms
```

Figure 2.20. Verify Connectivity by Hostname

2.4.2. Configure SSH

SSH is a convenient and secure way to copy files and perform commands remotely. For the purposes of this guide, we will create a key without a password (using the -N “” option) so that we can perform remote actions without being prompted.

Warning

Unprotected SSH keys, those without a password, are not recommended for servers exposed to the outside world.

Create a new key and allow anyone with that key to log in:

```
[root@pcmk-1 ~]# ssh-keygen -t dsa -f ~/.ssh/id_dsa -N ""
Generating public/private dsa key pair.
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
91:09:5c:82:5a:6a:50:08:4e:b2:0c:62:de:cc:74:44 root@pcmk-1.clusterlabs.org

The key's randomart image is:
+--[ DSA 1024]----+
|==.ooEo..        |
|X O + .o o       |
| * A    +        |
|  +      .       |
| .      S        |
|                 |
|                 |
|                 |
|                 |
+-----------------+
[root@pcmk-1 ~]# cp .ssh/id_dsa.pub .ssh/authorized_keys
[root@pcmk-1 ~]#
```

Figure 2.21. Creating and Activating a new SSH Key

Install the key on the other nodes and test that you can now run commands remotely, without being prompted

```
[root@pcmk-1 ~]# scp -r .ssh pcmk-2:
The authenticity of host 'pcmk-2 (192.168.122.102)' can't be established.
RSA key fingerprint is b1:2b:55:93:f1:d9:52:2b:0f:f2:8a:4e:ae:c6:7c:9a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pcmk-2,192.168.122.102' (RSA) to the list of known hosts.
root@pcmk-2's password:
id_dsa.pub                                100%  616     0.6KB/s   00:00
id_dsa                                    100%  672     0.7KB/s   00:00
known_hosts                               100%  400     0.4KB/s   00:00
authorized_keys                           100%  616     0.6KB/s   00:00
[root@pcmk-1 ~]# ssh pcmk-2 -- uname -n
pcmk-2
[root@pcmk-1 ~]#
```

Figure 2.22. Installing the SSH Key on Another Host

2.4.3. Short Node Names

During installation, we filled in the machine's fully qualifier domain name (FQDN) which can be rather long when it appears in cluster logs and status output. See for yourself how the machine identifies itself:

```
[root@pcmk-1 ~]# uname -n
pcmk-1.clusterlabs.org
[root@pcmk-1 ~]# dnsdomainname
clusterlabs.org
```

The output from the second command is fine, but we really don't need the domain name included in the basic host details. To address this, we need to update /etc/sysconfig/network. This is what it should look like before we start.

```
[root@pcmk-1 ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=pcmk-1.clusterlabs.org
GATEWAY=192.168.122.1
```

All we need to do now is strip off the domain name portion, which is stored elsewhere anyway.

```
[root@pcmk-1 ~]# sed -i.bak 's/\.[a-z].*//g' /etc/sysconfig/network
```

Now confirm the change was successful. The revised file contents should look something like this.

```
[root@pcmk-1 ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=pcmk-1
GATEWAY=192.168.122.1
```

However we're not finished. The machine wont normally see the shortened host name until about it reboots, but we can force it to update.

```
[root@pcmk-1 ~]# source /etc/sysconfig/network
[root@pcmk-1 ~]# hostname $HOSTNAME
```

Now check the machine is using the correct names

```
[root@pcmk-1 ~]# uname -n
pcmk-1
[root@pcmk-1 ~]# dnsdomainname
clusterlabs.org
```

Now repeat on pcmk-2.

2.4.4. Configuring Corosync

Choose a port number and multi-cast [7] address. [8]

Be sure that the values you chose do not conflict with any existing clusters you might have. For advice on choosing a multi-cast address, see *http://www.29west.com/docs/THPM/multicast-address-assignment.html* [9]

For this document, I have chosen port 4000 and used 226.94.1.1 as the multi-cast address.

```
[root@pcmk-1 ~]# export ais_port=4000
[root@pcmk-1 ~]# export ais_mcast=226.94.1.1
```

Next we automatically determine the hosts address. By not using the full address, we make the configuration suitable to be copied to other nodes.

[7] *http://en.wikipedia.org/wiki/Multicast*
[8] *http://en.wikipedia.org/wiki/Multicast_address*
[9] http://www.29west.com/docs/THPM/multicast-address-assignment.html

```
[root@pcmk-1 ~]# export ais_addr=`ip addr | grep "inet " | tail -n 1 | awk '{print $4}' | sed
 s/255/0/`
```

Display and verify the configuration options

```
[root@pcmk-1 ~]# env | grep ais_
ais_mcast=226.94.1.1
ais_port=4000
ais_addr=192.168.122.0
```

Once you're happy with the chosen values, update the Corosync configuration

```
[root@pcmk-1 ~]# cp /etc/corosync/corosync.conf.example /etc/corosync/corosync.conf
[root@pcmk-1 ~]# sed -i.bak "s/.*mcastaddr:.*/mcastaddr:\ $ais_mcast/g" /etc/corosync/
corosync.conf
[root@pcmk-1 ~]# sed -i.bak "s/.*mcastport:.*/mcastport:\ $ais_port/g" /etc/corosync/
corosync.conf
[root@pcmk-1 ~]# sed -i.bak "s/.*bindnetaddr:.*/bindnetaddr:\ $ais_addr/g" /etc/corosync/
corosync.conf
```

Finally, tell Corosync to start Pacemaker

```
[root@pcmk-1 ~]# cat <<-END >>/etc/corosync/service.d/pcmk
service {
        # Load the Pacemaker Cluster Resource Manager
        name: pacemaker
        ver:  0
}
END
```

The final configuration should look something like the sample in the appendix.

2.4.5. Propagate the Configuration

Now we need to copy the changes so far to the other node:

```
[root@pcmk-1 ~]# for f in /etc/corosync/corosync.conf /etc/corosync/service.d/pcmk /etc/hosts;
 do scp $f pcmk-2:$f ; done
corosync.conf                                 100% 1528     1.5KB/s   00:00
hosts                                         100%  281     0.3KB/s   00:00
[root@pcmk-1 ~]#
```

Verify Cluster Installation

3.1. Verify Corosync Installation

Start Corosync on the first node

```
[root@pcmk-1 ~]# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
```

Check the cluster started correctly and that an initial membership was able to form

```
[root@pcmk-1 ~]# grep -e "corosync.*network interface" -e "Corosync Cluster Engine" -e
 "Successfully read main configuration file" /var/log/messages
Aug 27 09:05:34 pcmk-1 corosync[1540]: [MAIN  ] Corosync Cluster Engine ('1.1.0'): started and
 ready to provide service.
Aug 27 09:05:34 pcmk-1 corosync[1540]: [MAIN  ] Successfully read main configuration file '/
etc/corosync/corosync.conf'.
[root@pcmk-1 ~]# grep TOTEM /var/log/messages
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transport (UDP/IP).
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transmit/receive security:
 libtomcrypt SOBER128/SHA1HMAC (mode 0).
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] The network interface [192.168.122.101] is now
 up.
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] A processor joined or left the membership and
 a new membership was formed.
```

With one node functional, its now safe to start Corosync on the second node as well.

```
[root@pcmk-1 ~]# ssh pcmk-2 -- /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
[root@pcmk-1 ~]#
```

Check the cluster formed correctly

```
[root@pcmk-1 ~]# grep TOTEM /var/log/messages
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transport (UDP/IP).
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transmit/receive security:
 libtomcrypt SOBER128/SHA1HMAC (mode 0).
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] The network interface [192.168.122.101] is now
 up.
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] A processor joined or left the membership and
 a new membership was formed.
Aug 27 09:12:11 pcmk-1 corosync[1540]: [TOTEM ] A processor joined or left the membership and
 a new membership was formed.
```

3.2. Verify Pacemaker Installation

Now that we have confirmed that Corosync is functional we can check the rest of the stack.

```
[root@pcmk-1 ~]# grep pcmk_startup /var/log/messages
```

```
Aug 27 09:05:35 pcmk-1 corosync[1540]:   [pcmk  ] info: pcmk_startup: CRM: Initialized
Aug 27 09:05:35 pcmk-1 corosync[1540]:   [pcmk  ] Logging: Initialized pcmk_startup
Aug 27 09:05:35 pcmk-1 corosync[1540]:   [pcmk  ] info: pcmk_startup: Maximum core file size
 is: 18446744073709551615
Aug 27 09:05:35 pcmk-1 corosync[1540]:   [pcmk  ] info: pcmk_startup: Service: 9
Aug 27 09:05:35 pcmk-1 corosync[1540]:   [pcmk  ] info: pcmk_startup: Local hostname: pcmk-1
```

Now verify the Pacemaker processes have been started

```
[root@pcmk-1 ~]# ps axf
  PID TTY      STAT   TIME COMMAND
    2 ?        S<     0:00 [kthreadd]
    3 ?        S<     0:00  \_ [migration/0]
... lots of processes ...
 2166 pts/0    SLl    0:01 /usr/sbin/corosync
 2172 ?        SLs    0:00  \_ /usr/lib64/heartbeat/stonithd
 2173 pts/0    S      0:00  \_ /usr/lib64/heartbeat/cib
 2174 pts/0    S      0:00  \_ /usr/lib64/heartbeat/lrmd
 2175 pts/0    S      0:00  \_ /usr/lib64/heartbeat/attrd
 2176 pts/0    S      0:00  \_ /usr/lib64/heartbeat/pengine
 2177 pts/0    S      0:00  \_ /usr/lib64/heartbeat/crmd
```

And finally, check for any ERRORs during startup, there shouldn't be any, and display the cluster's status.

```
[root@pcmk-1 ~]# grep ERROR: /var/log/messages | grep -v unpack_resources
[root@pcmk-1 ~]# crm_mon
============
Last updated: Thu Aug 27 16:54:55 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
0 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]
```

Using Pacemaker Tools

In the dark past, configuring Pacemaker required the administrator to read and write XML. In true UNIX style, there were also a number of different commands that specialized in different aspects of querying and updating the cluster.

Since Pacemaker 1.0, this has all changed and we have an integrated, scriptable, cluster shell that hides all the messy XML scaffolding. It even allows you to queue up several changes at once and commit them atomically.

Take some time to familiarize yourself with what it can do.

```
[root@pcmk-1 ~]# crm --help

usage:
    crm [-D display_type]
    crm [-D display_type] args
    crm [-D display_type] [-f file]

    Use crm without arguments for an interactive session.
    Supply one or more arguments for a "single-shot" use.
    Specify with -f a file which contains a script. Use '-' for
    standard input or use pipe/redirection.

    crm displays cli format configurations using a color scheme
    and/or in uppercase. Pick one of "color" or "uppercase", or
    use "-D color,uppercase" if you want colorful uppercase.
    Get plain output by "-D plain". The default may be set in
    user preferences (options).

Examples:

    # crm -f stopapp2.cli
    # crm < stopapp2.cli
    # crm resource stop global_www
    # crm status
```

The primary tool for monitoring the status of the cluster is crm_mon (also available as crm status). It can be run in a variety of modes and has a number of output options. To find out about any of the tools that come with Pacemaker, simply invoke them with the **--help** option or consult the included man pages. Both sets of output are created from the tool, and so will always be in sync with each other and the tool itself.

Additionally, the Pacemaker version and supported cluster stack(s) is available via the **--version** option.

```
[root@pcmk-1 ~]# crm_mon --version
crm_mon 1.0.5 for OpenAIS and Heartbeat (Build: 462f1569a43740667daf7b0f6b521742e9eb8fa7)

Written by Andrew Beekhof
[root@pcmk-1 ~]# crm_mon --help
crm_mon - Provides a summary of cluster's current state.

Outputs varying levels of detail in a number of different formats.

Usage: crm_mon mode [options]
Options:
```

```
 -?, --help                  This text
 -$, --version               Version information
 -V, --verbose               Increase debug output

Modes:
 -h, --as-html=value         Write cluster status to the named file
 -w, --web-cgi               Web mode with output suitable for cgi
 -s, --simple-status         Display the cluster status once as a simple one line output
 (suitable for nagios)
 -S, --snmp-traps=value      Send SNMP traps to this station
 -T, --mail-to=value         Send Mail alerts to this user.  See also --mail-from, --mail-host,
 --mail-prefix

Display Options:
 -n, --group-by-node         Group resources by node
 -r, --inactive              Display inactive resources
 -f, --failcounts            Display resource fail counts
 -o, --operations            Display resource operation history
 -t, --timing-details        Display resource operation history with timing details


Additional Options:
 -i, --interval=value           Update frequency in seconds
 -1, --one-shot                 Display the cluster status once on the console and exit
 -N, --disable-ncurses          Disable the use of ncurses
 -d, --daemonize                Run in the background as a daemon
 -p, --pid-file=value           (Advanced) Daemon pid file location
 -F, --mail-from=value          Mail alerts should come from the named user
 -H, --mail-host=value          Mail alerts should be sent via the named host
 -P, --mail-prefix=value        Subjects for mail alerts should start with this string
 -E, --external-agent=value     A program to run when resource operations take place.
 -e, --external-recipient=value A recipient for your program (assuming you want the program to
 send something to someone).

Examples:

Display the cluster´s status on the console with updates as they occur:
        # crm_mon

Display the cluster´s status on the console just once then exit:
        # crm_mon

Display your cluster´s status, group resources by node, and include inactive resources in the
 list:
        # crm_mon --group-by-node --inactive

Start crm_mon as a background daemon and have it write the cluster´s status to an HTML file:
        # crm_mon --daemonize --as-html /path/to/docroot/filename.html

Start crm_mon as a background daemon and have it send email alerts:
        # crm_mon --daemonize --mail-to user@example.com --mail-host mail.example.com

Start crm_mon as a background daemon and have it send SNMP alerts:
        # crm_mon --daemonize --snmp-traps snmptrapd.example.com

Report bugs to pacemaker@oss.clusterlabs.org
```

Note

If the SNMP and/or email options are not listed, then Pacemaker was not built to support them. This may be by the choice of your distribution or the required libraries may not have been available. Please contact whoever supplied you with the packages for more details.

Creating an Active/Passive Cluster

5.1. Exploring the Existing Configuration

When Pacemaker starts up, it automatically records the number and details of the nodes in the cluster as well as which stack is being used and the version of Pacemaker being used.

This is what the base configuration should look like.

```
[root@pcmk-2 ~]# crm configure show
node pcmk-1
node pcmk-2
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2"
```

For those that are not of afraid of XML, you can see the raw configuration by appending “xml” to the previous command.

```
[root@pcmk-2 ~]# crm configure show xml
<?xml version="1.0" ?>
<cib admin_epoch="0" crm_feature_set="3.0.1" dc-uuid="pcmk-1" epoch="13" have-quorum="1"
 num_updates="7" validate-with="pacemaker-1.0">
  <configuration>
    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <nvpair id="cib-bootstrap-options-dc-version" name="dc-version"
 value="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7"/>
        <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-
infrastructure" value="openais"/>
        <nvpair id="cib-bootstrap-options-expected-quorum-votes" name="expected-quorum-votes"
 value="2"/>
      </cluster_property_set>
    </crm_config>
    <rsc_defaults/>
    <op_defaults/>
    <nodes>
      <node id="pcmk-1" type="normal" uname="pcmk-1"/>
      <node id="pcmk-2" type="normal" uname="pcmk-2"/>
    </nodes>
    <resources/>
    <constraints/>
  </configuration>
</cib>
```

The last XML you’ll see in this document

Before we make any changes, its a good idea to check the validity of the configuration.

```
[root@pcmk-1 ~]# crm_verify -L
crm_verify[2195]: 2009/08/27_16:57:12 ERROR: unpack_resources: Resource start-up disabled
 since no STONITH resources have been defined
crm_verify[2195]: 2009/08/27_16:57:12 ERROR: unpack_resources: Either configure some or
 disable STONITH with the stonith-enabled option
```

```
crm_verify[2195]: 2009/08/27_16:57:12 ERROR: unpack_resources: NOTE: Clusters with shared data
 need STONITH to ensure data integrity
Errors found during check: config not valid
  -V may provide more details
[root@pcmk-1 ~]#
```

As you can see, the tool has found some errors.

In order to guarantee the safety of your data [1] , Pacemaker ships with STONITH [2] enabled. However it also knows when no STONITH configuration has been supplied and reports this as a problem (since the cluster would not be able to make progress if a situation requiring node fencing arose).

For now, we will disable this feature and configure it later in the Configuring STONITH section. It is important to note that the use of STONITH is highly encouraged, turning it off tells the cluster to simply pretend that failed nodes are safely powered off. Some vendors will even refuse to support clusters that have it disabled.

To disable STONITH, we set the stonith-enabled cluster option to false.

```
crm configure property stonith-enabled=false

crm_verify -L
```

With the new cluster option set, the configuration is now valid.

5.2. Adding a Resource

The first thing we should do is configure an IP address. Regardless of where the cluster service(s) are running, we need a consistent address to contact them on. Here I will choose and add 192.168.122.101 as the floating address, give it the imaginative name ClusterIP and tell the cluster to check that its running every 30 seconds.

Important

The chosen address must not be one already associated with a physical node

```
crm configure primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip=192.168.122.101 cidr_netmask=32 \
        op monitor interval=30s
```

The other important piece of information here is ocf:heartbeat:IPaddr2. This tells Pacemaker three things about the resource you want to add. The first field, ocf, is the standard to which the resource script conforms to and where to find it. The second field is specific to OCF resources and tells the cluster which namespace to find the resource script in, in this case heartbeat. The last field indicates the name of the resource script.

To obtain a list of the available resource classes, run

```
[root@pcmk-1 ~]# crm ra classes
```

[1] If the data is corrupt, there is little point in continuing to make it available

[2] A common node fencing mechanism. Used to ensure data integrity by powering off “bad” nodes.

```
heartbeat
lsb
ocf / heartbeat pacemaker
stonith
```

To then find all the OCF resource agents provided by Pacemaker and Heartbeat, run

```
[root@pcmk-1 ~]# crm ra list ocf pacemaker
ClusterMon     Dummy          Stateful       SysInfo        SystemHealth   controld
ping           pingd
[root@pcmk-1 ~]# crm ra list ocf heartbeat
AoEtarget              AudibleAlarm           ClusterMon             Delay
Dummy                  EvmsSCC                Evmsd                  Filesystem
ICP                    IPaddr                 IPaddr2                IPsrcaddr
LVM                    LinuxSCSI              MailTo                 ManageRAID
ManageVE               Pure-FTPd              Raid1                  Route
SAPDatabase            SAPInstance            SendArp                ServeRAID
SphinxSearchDaemon     Squid                  Stateful               SysInfo
VIPArip                VirtualDomain          WAS                    WAS6
WinPopup               Xen                    Xinetd                 anything
apache                 db2                    drbd                   eDir88
iSCSILogicalUnit       iSCSITarget            ids                    iscsi
ldirectord             mysql                  mysql-proxy            nfsserver
oracle                 oralsnr                pgsql                  pingd
portblock              rsyncd                 scsi2reservation       sfex
tomcat                 vmware
[root@pcmk-1 ~]#
```

Now verify that the IP resource has been added and display the cluster's status to see that it is now active.

```
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive ClusterIP ocf:heartbeat:IPaddr2 \
 params ip="192.168.122.101" cidr_netmask="32" \
 op monitor interval="30s"
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 15:23:48 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]
ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-1
```

5.3. Perform a Failover

Being a high-availability cluster, we should test failover of our new resource before moving on.

First, find the node on which the IP address is running.

```
[root@pcmk-1 ~]# crm resource status ClusterIP
resource ClusterIP is running on: pcmk-1
[root@pcmk-1 ~]#
```

Shut down Corosync on that machine.

```
[root@pcmk-1 ~]# ssh pcmk-1 -- /etc/init.d/corosync stop
Stopping Corosync Cluster Engine (corosync): [ OK ]
Waiting for services to unload: [ OK ]
[root@pcmk-1 ~]#
```

Once Corosync is no longer running, go to the other node and check the cluster status with crm_mon.

```
[root@pcmk-2 ~]# crm_mon
============
Last updated: Fri Aug 28 15:27:35 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============

Online: [ pcmk-2 ]
OFFLINE: [ pcmk-1 ]
```

There are three things to notice about the cluster's current state. The first is that, as expected, pcmk-1 is now offline. However we can also see that ClusterIP isn't running anywhere!

5.3.1. Quorum and Two-Node Clusters

This is because the cluster no longer has quorum, as can be seen by the text "partition WITHOUT quorum" (emphasised green) in the output above. In order to reduce the possibility of data corruption, Pacemaker's default behavior is to stop all resources if the cluster does not have quorum.

A cluster is said to have quorum when more than half the known or expected nodes are online, or for the mathematically inclined, whenever the following equation is true:

total_nodes - 1 < 2 * active_nodes

Therefore a two-node cluster only has quorum when both nodes are running, which is no longer the case for our cluster. This would normally make the creation of a two-node cluster pointless[3], however it is possible to control how Pacemaker behaves when quorum is lost. In particular, we can tell the cluster to simply ignore quorum altogether.

```
[root@pcmk-1 ~]# crm configure property no-quorum-policy=ignore
[root@pcmk-1 ~]# crm configure show
node pcmk-1
```

[3] Actually some would argue that two-node clusters are always pointless, but that is an argument for another time.

```
node pcmk-2
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
```

After a few moments, the cluster will start the IP address on the remaining node. Note that the cluster still does not have quorum.

```
[root@pcmk-2 ~]# crm_mon
============
Last updated: Fri Aug 28 15:30:18 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============
Online: [ pcmk-2 ]
OFFLINE: [ pcmk-1 ]

ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-2
```

Now simulate node recovery by restarting the cluster stack on pcmk-1 and check the cluster's status.

```
[root@pcmk-1 ~]# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 15:32:13 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============
Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-1
```

Here we see something that some may consider surprising, the IP is back running at its original location!

5.3.2. Prevent Resources from Moving after Recovery

In some circumstances it is highly desirable to prevent healthy resources from being moved around the cluster. Move resources almost always requires a period of downtime and for complex services like Oracle databases, this period can be quite long.

To address this, Pacemaker has the concept of resource stickiness which controls how much a service prefers to stay running where it is. You may like to think of it as the "cost" of any downtime. By default, Pacemaker assumes there is zero cost associated with moving resources and will do so to

achieve “optimal[4]” resource placement. We can specify a different stickiness for every resource, but it is often sufficient to change the default.

```
crm configure rsc_defaults resource-stickiness=100
[root@pcmk-2 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
 resource-stickiness="100"
```

If we now retry the failover test, we see that as expected ClusterIP still moves to pcmk-2 when pcmk-1 is taken offline.

```
[root@pcmk-1 ~]# ssh pcmk-1 -- /etc/init.d/corosync stop
Stopping Corosync Cluster Engine (corosync):               [  OK  ]
Waiting for services to unload:                            [  OK  ]
[root@pcmk-1 ~]# ssh pcmk-2 -- crm_mon -1
============
Last updated: Fri Aug 28 15:39:38 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============

Online: [ pcmk-2 ]
OFFLINE: [ pcmk-1 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
```

However when we bring pcmk-1 back online, ClusterIP now remains running on pcmk-2.

```
[root@pcmk-1 ~]# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 15:41:23 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============
```

[4] It should be noted that Pacemaker’s definition of optimal may not always agree with that of a human’s. The order in which Pacemaker processes lists of resources and nodes create implicit preferences (required in order to create a stabile solution) in situations where the administrator had not explicitly specified some.

```
Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
```

Apache - Adding More Services

Note

Now that we have a basic but functional active/passive two-node cluster, we're ready to add some real services. We're going to start with Apache because its a feature of many clusters and relatively simple to configure.

6.1. Installation

Before continuing, we need to make sure Apache is installed on *both* hosts.

```
[root@ppcmk-1 ~]# yum install -y httpd
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.2.13-2.fc12 set to be updated
--> Processing Dependency: httpd-tools = 2.2.13-2.fc12 for package: httpd-2.2.13-2.fc12.x86_64
--> Processing Dependency: apr-util-ldap for package: httpd-2.2.13-2.fc12.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.2.13-2.fc12.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.2.13-2.fc12.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.2.13-2.fc12.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.3.9-2.fc12 set to be updated
---> Package apr-util.x86_64 0:1.3.9-2.fc12 set to be updated
---> Package apr-util-ldap.x86_64 0:1.3.9-2.fc12 set to be updated
---> Package httpd-tools.x86_64 0:2.2.13-2.fc12 set to be updated
---> Package mailcap.noarch 0:2.1.30-1.fc12 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package               Arch             Version                Repository         Size
================================================================================
Installing:
 httpd                 x86_64           2.2.13-2.fc12          rawhide           735 k
Installing for dependencies:
 apr                   x86_64           1.3.9-2.fc12           rawhide           117 k
 apr-util              x86_64           1.3.9-2.fc12           rawhide            84 k
 apr-util-ldap         x86_64           1.3.9-2.fc12           rawhide            15 k
 httpd-tools           x86_64           2.2.13-2.fc12          rawhide            63 k
 mailcap               noarch           2.1.30-1.fc12          rawhide            25 k

Transaction Summary
================================================================================
Install       6 Package(s)
Upgrade       0 Package(s)

Total download size: 1.0 M
Downloading Packages:
(1/6): apr-1.3.9-2.fc12.x86_64.rpm                                | 117 kB     00:00
(2/6): apr-util-1.3.9-2.fc12.x86_64.rpm                           |  84 kB     00:00
(3/6): apr-util-ldap-1.3.9-2.fc12.x86_64.rpm                      |  15 kB     00:00
(4/6): httpd-2.2.13-2.fc12.x86_64.rpm                             | 735 kB     00:00
(5/6): httpd-tools-2.2.13-2.fc12.x86_64.rpm                       |  63 kB     00:00
(6/6): mailcap-2.1.30-1.fc12.noarch.rpm                           |  25 kB     00:00
```

```
--------------------------------------------------------------------------------
Total                                                  875 kB/s | 1.0 MB     00:01
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : apr-1.3.9-2.fc12.x86_64                                     1/6
  Installing     : apr-util-1.3.9-2.fc12.x86_64                                2/6
  Installing     : apr-util-ldap-1.3.9-2.fc12.x86_64                           3/6
  Installing     : httpd-tools-2.2.13-2.fc12.x86_64                            4/6
  Installing     : mailcap-2.1.30-1.fc12.noarch                                5/6
  Installing     : httpd-2.2.13-2.fc12.x86_64                                  6/6

Installed:
  httpd.x86_64 0:2.2.13-2.fc12

Dependency Installed:
  apr.x86_64 0:1.3.9-2.fc12            apr-util.x86_64 0:1.3.9-2.fc12
  apr-util-ldap.x86_64 0:1.3.9-2.fc12  httpd-tools.x86_64 0:2.2.13-2.fc12
  mailcap.noarch 0:2.1.30-1.fc12

Complete!
[root@pcmk-1 ~]#
```

Also, we need the wget tool in order for the cluster to be able to check the status of the Apache server.

```
[root@pcmk-1 ~]# yum install -y wget
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.11.4-5.fc12 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package        Arch             Version                     Repository      Size
================================================================================
Installing:
 wget           x86_64           1.11.4-5.fc12               rawhide        393 k

Transaction Summary
================================================================================
Install       1 Package(s)
Upgrade       0 Package(s)

Total download size: 393 k
Downloading Packages:
wget-1.11.4-5.fc12.x86_64.rpm                                | 393 kB     00:00

Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : wget-1.11.4-5.fc12.x86_64                                   1/1

Installed:
  wget.x86_64 0:1.11.4-5.fc12

Complete!
```

```
[root@pcmk-1 ~]#
```

6.2. Preparation

First we need to create a page for Apache to serve up. On Fedora the default Apache docroot is /var/www/html, so we'll create an index file there.

```
[root@pcmk-1 ~]# cat <<-END >/var/www/html/index.html
 <html>
 <body>My Test Site - pcmk-1</body>
 </html>
 END
[root@pcmk-1 ~]#
```

For the moment, we will simplify things by serving up only a static site and manually sync the data between the two nodes. So run the command again on pcmk-2.

```
[root@pcmk-2 ~]# cat <<-END >/var/www/html/index.html
 <html>
 <body>My Test Site - pcmk-2</body>
 </html>
 END
[root@pcmk-2 ~]#
```

6.3. Update the Configuration

At this point, Apache is ready to go, all that needs to be done is to add it to the cluster. Lets call the resource WebSite. We need to use an OCF script called apache in the heartbeat namespace [1] , the only required parameter is the path to the main Apache configuration file and we'll tell the cluster to check once a minute that apache is still running.

```
[root@pcmk-1 ~]# crm configure primitive WebSite ocf:heartbeat:apache params configfile=/etc/
httpd/conf/httpd.conf op monitor interval=1min
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
 params configfile="/etc/httpd/conf/httpd.conf" \
 op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
```

After a short delay, we should see the cluster start apache

[1] Compare the key used here ocf:heartbeat:apache with the one we used earlier for the IP address: ocf:heartbeat:IPaddr2

```
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 16:12:49 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
WebSite        (ocf::heartbeat:apache):        Started pcmk-1
```

Wait a moment, the WebSite resource isn't running on the same host as our IP address!

6.4. Ensuring Resources Run on the Same Host

To reduce the load on any one machine, Pacemaker will generally try to spread the configured resources across the cluster nodes. However we can tell the cluster that two resources are related and need to run on the same host (or not at all). Here we instruct the cluster that WebSite can only run on the host that ClusterIP is active on. If ClusterIP is not active anywhere, WebSite will not be permitted to run anywhere.

```
[root@pcmk-1 ~]# crm configure colocation website-with-ip INFINITY: WebSite ClusterIP
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
colocation website-with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 16:14:34 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
WebSite        (ocf::heartbeat:apache):        Started pcmk-2
```

6.5. Controlling Resource Start/Stop Ordering

When Apache starts, it binds to the available IP addresses. It doesn't know about any addresses we add afterwards, so not only do they need to run on the same node, but we need to make sure ClusterIP is already active before we start WebSite. We do this by adding an ordering constraint. We need to give it a name (chose something descriptive like apache-after-ip), indicate that its mandatory (so that any recovery for ClusterIP will also trigger recovery of WebSite) and list the two resources in the order we need them to start.

```
[root@pcmk-1 ~]# crm configure order apache-after-ip mandatory: ClusterIP WebSite
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
colocation website-with-ip inf: WebSite ClusterIP
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
```

6.6. Specifying a Preferred Location

Pacemaker does not rely on any sort of hardware symmetry between nodes, so it may well be that one machine is more powerful than the other. In such cases it makes sense to host the resources there if it is available. To do this we create a location constraint. Again we give it a descriptive name (prefer-pcmk-1), specify the resource we want to run there (WebSite), how badly we'd like it to run there (we'll use 50 for now, but in a two-node situation almost any value above 0 will do) and the host's name.

```
[root@pcmk-1 ~]# crm configure location prefer-pcmk-1 WebSite 50: pcmk-1
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation website-with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
```

```
        resource-stickiness="100"
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 16:17:35 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
WebSite        (ocf::heartbeat:apache):        Started pcmk-2
```

Wait a minute, the resources are still on pcmk-2!

Even though we now prefer pcmk-1 over pcmk-2, that preference is (intentionally) less than the resource stickiness (how much we preferred not to have unnecessary downtime).

To see the current placement scores, you can use a tool called ptest

ptest -sL

Note

Include output

There is a way to force them to move though...

6.7. Manually Moving Resources Around the Cluster

There are always times when an administrator needs to override the cluster and force resources to move to a specific location. Underneath we use location constraints like the one we created above, happily you don't need to care. Just provide the name of the resource and the intended location, we'll do the rest.

```
[root@pcmk-1 ~]# crm resource move WebSite pcmk-1
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 16:19:24 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-1
WebSite        (ocf::heartbeat:apache):        Started pcmk-1
Notice how the colocation rule we created has ensured that ClusterIP was also moved to pcmk-1.
For the curious, we can see the effect of this command by examining the configuration
crm configure show
```

```
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
location cli-prefer-WebSite WebSite \
 rule $id="cli-prefer-rule-WebSite" inf: #uname eq pcmk-1
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation website-with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
```

Highlighted is the automated constraint used to move the resources to pcmk-1

6.7.1. Giving Control Back to the Cluster

Once we've finished whatever activity that required us to move the resources to pcmk-1, in our case nothing, we can then allow the cluster to resume normal operation with the unmove command. Since we previously configured a default stickiness, the resources will remain on pcmk-1.

```
[root@pcmk-1 ~]# crm resource unmove WebSite
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation website-with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
```

Note that the automated constraint is now gone. If we check the cluster status, we can also see that as expected the resources are still active on pcmk-1.

```
[root@pcmk-1 ~]# crm_mon
============
Last updated: Fri Aug 28 16:20:53 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
```

```
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

 ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-1
 WebSite        (ocf::heartbeat:apache):        Started pcmk-1
```

Replicated Storage with DRBD

Even if you're serving up static websites, having to manually synchronize the contents of that website to all the machines in the cluster is not ideal. For dynamic websites, such as a wiki, its not even an option. Not everyone care afford network-attached storage but somehow the data needs to be kept in sync. Enter DRBD which can be thought of as network based RAID-1. See *http://www.drbd.org*[1]/ for more details.

7.1. Install the DRBD Packages

Since its inclusion in the upstream 2.6.33 kernel, everything needed to use DRBD ships with Fedora 13. All you need to do is install it:

```
[root@pcmk-1 ~]# yum install -y drbd-pacemaker
Loaded plugins: presto, refresh-packagekit
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package drbd-pacemaker.x86_64 0:8.3.7-2.fc13 set to be updated
--> Processing Dependency: drbd-utils = 8.3.7-2.fc13 for package: drbd-
pacemaker-8.3.7-2.fc13.x86_64
--> Running transaction check
---> Package drbd-utils.x86_64 0:8.3.7-2.fc13 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch           Version              Repository      Size
================================================================================
Installing:
 drbd-pacemaker         x86_64         8.3.7-2.fc13         fedora          19 k
Installing for dependencies:
 drbd-utils             x86_64         8.3.7-2.fc13         fedora         165 k

Transaction Summary
================================================================================
Install       2 Package(s)
Upgrade       0 Package(s)

Total download size: 184 k
Installed size: 427 k
Downloading Packages:
Setting up and reading Presto delta metadata
fedora/prestodelta                                       | 1.7 kB     00:00
Processing delta metadata
Package(s) data still to download: 184 k
(1/2): drbd-pacemaker-8.3.7-2.fc13.x86_64.rpm            |  19 kB     00:01
(2/2): drbd-utils-8.3.7-2.fc13.x86_64.rpm                | 165 kB     00:02
--------------------------------------------------------------------------------
Total                                            45 kB/s | 184 kB     00:04
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : drbd-utils-8.3.7-2.fc13.x86_64                          1/2
```

[1] http://www.drbd.org/

```
  Installing     : drbd-pacemaker-8.3.7-2.fc13.x86_64                          2/2

Installed:
  drbd-pacemaker.x86_64 0:8.3.7-2.fc13

Dependency Installed:
  drbd-utils.x86_64 0:8.3.7-2.fc13

Complete!
[root@pcmk-1 ~]#
```

7.2. Configure DRBD

Before we configure DRBD, we need to set aside some disk for it to use.

7.2.1. Create A Partition for DRBD

If you have more than 1Gb free, feel free to use it. For this guide however, 1Gb is plenty of space for a single html file and sufficient for later holding the GFS2 metadata.

```
[root@pcmk-1 ~]# lvcreate -n drbd-demo -L 1G VolGroup
  Logical volume "drbd-demo" created
[root@pcmk-1 ~]# lvs
  LV        VG       Attr   LSize   Origin Snap%  Move Log Copy%  Convert
  drbd-demo VolGroup -wi-a- 1.00G
  lv_root   VolGroup -wi-ao   7.30G
  lv_swap   VolGroup -wi-ao 500.00M
```

Repeat this on the second node, be sure to use the same size partition.

```
[root@pcmk-2 ~]# lvs
  LV      VG       Attr   LSize   Origin Snap%  Move Log Copy%  Convert
  lv_root VolGroup -wi-ao   7.30G
  lv_swap VolGroup -wi-ao 500.00M
[root@pcmk-2 ~]# lvcreate -n drbd-demo -L 1G VolGroup
  Logical volume "drbd-demo" created
[root@pcmk-2 ~]# lvs
  LV        VG       Attr   LSize   Origin Snap%  Move Log Copy%  Convert
  drbd-demo VolGroup -wi-a- 1.00G
  lv_root   VolGroup -wi-ao   7.30G
  lv_swap   VolGroup -wi-ao 500.00M
```

7.2.2. Write the DRBD Config

There is no series of commands for build a DRBD configuration, so simply copy the configuration below to /etc/drbd.conf

Detailed information on the directives used in this configuration (and other alternatives) is available from *http://www.drbd.org/users-guide/ch-configure.html*

Warning

Be sure to use the names and addresses of *your* nodes if they differ from the ones used in this guide.

```
global {
  usage-count yes;
}
common {
  protocol C;
}
resource wwwdata {
  meta-disk internal;
  device    /dev/drbd1;
  syncer {
    verify-alg sha1;
  }
  net {
    allow-two-primaries;
  }
  on pcmk-1 {
    disk      /dev/mapper/VolGroup-drbd--demo;
    address   192.168.122.101:7789;
  }
  on
pcmk-2 {
    disk      /dev/mapper/VolGroup-drbd--demo;
    address   192.168.122.102:7789;
  }
}
```

Note

TODO: Explain the reason for the allow-two-primaries option

7.2.3. Initialize and Load DRBD

With the configuration in place, we can now perform the DRBD initialization

```
[root@pcmk-1 ~]# drbdadm create-md wwwdata
md_offset 12578816
al_offset 12546048
bm_offset 12541952

Found some data
 ==> This might destroy existing data! <==

Do you want to proceed?
[need to type 'yes' to confirm] yes

Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
```

Now load the DRBD kernel module and confirm that everything is sane

```
[root@pcmk-1 ~]# modprobe drbd
[root@pcmk-1 ~]# drbdadm up wwwdata
```

```
[root@pcmk-1 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57

 1: cs:WFConnection ro:Secondary/Unknown ds:Inconsistent/DUnknown C r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:12248
[root@pcmk-1 ~]#

Repeat on the second node
drbdadm --force create-md wwwdata
modprobe drbd
drbdadm up wwwdata
cat /proc/drbd
[root@pcmk-2 ~]# drbdadm --force create-md wwwdata
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
[root@pcmk-2 ~]# modprobe drbd
WARNING: Deprecated config file /etc/modprobe.conf, all config files belong into /etc/
modprobe.d/.
[root@pcmk-2 ~]# drbdadm up wwwdata
[root@pcmk-2 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57

 1: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:12248
```

Now we need to tell DRBD which set of data to use. Since both sides contain garbage, we can run the following on pcmk-1:

```
[root@pcmk-1 ~]# drbdadm -- --overwrite-data-of-peer primary wwwdata
[root@pcmk-1 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57
 1: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r----
    ns:2184 nr:0 dw:0 dr:2472 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:10064
        [=====>..............] sync'ed: 33.4% (10064/12248)K
        finish: 0:00:37 speed: 240 (240) K/sec
[root@pcmk-1 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57
 1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:12248 nr:0 dw:0 dr:12536 al:0 bm:1 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0
```

pcmk-1 is now in the Primary state which allows it to be written to. Which means its a good point at which to create a filesystem and populate it with some data to serve up via our WebSite resource.

7.2.4. Populate DRBD with Data

```
[root@pcmk-1 ~]# mkfs.ext4 /dev/drbd1
mke2fs 1.41.4 (27-Jan-2009)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
3072 inodes, 12248 blocks
```

```
612 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=12582912
2 block groups
8192 blocks per group, 8192 fragments per group
1536 inodes per group
Superblock backups stored on blocks:
        8193

Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.

Now mount the newly created filesystem so we can create our index file
mount /dev/drbd1 /mnt/
cat <<-END >/mnt/index.html
<html>
<body>My Test Site - drbd</body>
</html>
END
umount /dev/drbd1
[root@pcmk-1 ~]# mount /dev/drbd1 /mnt/
[root@pcmk-1 ~]# cat <<-END >/mnt/index.html
> <html>
> <body>My Test Site - drbd</body>
> </html>
> END
[root@pcmk-1 ~]# umount /dev/drbd1
```

7.3. Configure the Cluster for DRBD

One handy feature of the crm shell is that you can use it in interactive mode to make several changes atomically.

First we launch the shell. The prompt will change to indicate you're in interactive mode.

```
[root@pcmk-1 ~]# crm
cib crm(live)#
```

Next we must create a working copy or the current configuration. This is where all our changes will go. The cluster will not see any of them until we say its ok. Notice again how the prompt changes, this time to indicate that we're no longer looking at the live cluster.

```
cib crm(live)# cib new drbd
INFO: drbd shadow CIB created
crm(drbd)#
```

Now we can create our DRBD clone and display the revised configuration.

```
crm(drbd)# configure primitive ocf:linbit:drbd WebData params drbd_resource=wwwdata \
 op monitor interval=60s
crm(drbd)# configure ms WebDataClone WebData meta master-max=1 master-node-max=1 \
 clone-max=2 clone-node-max=1 notify=true
```

```
crm(drbd)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
 params drbd_resource="wwwdata" \
 op monitor interval="60s"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
ms WebDataClone WebData \
 meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation website-with-ip inf: WebSite ClusterIP
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

Once we’re happy with the changes, we can tell the cluster to start using them and use crm_mon to check everything is functioning.

```
crm(drbd)# cib commit drbd
INFO: commited 'drbd' shadow CIB to the cluster
crm(drbd)# quit
bye
[root@pcmk-1 ~]# crm_mon
============
Last updated: Tue Sep  1 09:37:13 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
3 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-1
WebSite (ocf::heartbeat:apache):        Started pcmk-1
Master/Slave Set: WebDataClone
 Masters: [ pcmk-2 ]
 Slaves: [ pcmk-1 ]
```

Note

Include details on adding a second DRBD resource

Now that DRBD is functioning we can configure a Filesystem resource to use it. In addition to the filesystem’s definition, we also need to tell the cluster where it can be located (only on the DRBD Primary) and when it is allowed to start (after the Primary was promoted).

Once again we'll use the shell's interactive mode

```
[root@pcmk-1 ~]# crm
crm(live)# cib new fs
INFO: fs shadow CIB created
crm(fs)# configure primitive WebFS ocf:heartbeat:Filesystem \
 params device="/dev/mapper/VolGroup-drbd--demo" directory="/var/www/html" fstype="ext4"
crm(fs)# configure colocation fs_on_drbd inf: WebFS WebDataClone:Master
crm(fs)# configure order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
```

We also need to tell the cluster that Apache needs to run on the same machine as the filesystem and that it must be active before Apache can start.

```
crm(fs)# configure colocation WebSite-with-WebFS inf: WebSite WebFS
crm(fs)# configure order WebSite-after-WebFS inf: WebFS WebSite
```

Time to review the updated configuration:

```
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="ext4"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
ms WebDataClone WebData \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

After reviewing the new configuration, we again upload it and watch the cluster put it into effect.

```
crm(fs)# cib commit fs
INFO: commited 'fs' shadow CIB to the cluster
crm(fs)# quit
bye
```

```
[root@pcmk-1 ~]# crm_mon
============
Last updated: Tue Sep  1 10:08:44 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
4 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-1
WebSite (ocf::heartbeat:apache): Started pcmk-1
Master/Slave Set: WebDataClone
        Masters: [ pcmk-1 ]
        Slaves: [ pcmk-2 ]
WebFS (ocf::heartbeat:Filesystem): Started pcmk-1
```

7.3.1. Testing Migration

We could shut down the active node again, but another way to safely simulate recovery is to put the node into what is called “standby mode”. Nodes in this state tell the cluster that they are not allowed to run resources. Any resources found active there will be moved elsewhere. This feature can be particularly useful when updating the resources’ packages.

Put the local node into standby mode and observe the cluster move all the resources to the other node. Note also that the node’s status will change to indicate that it can no longer host resources.

```
[root@pcmk-1 ~]# crm node standby
[root@pcmk-1 ~]# crm_mon
============
Last updated: Tue Sep  1 10:09:57 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
4 Resources configured.
============

Node pcmk-1: standby
Online: [ pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
WebSite (ocf::heartbeat:apache):        Started pcmk-2
Master/Slave Set: WebDataClone
        Masters: [ pcmk-2 ]
        Stopped: [ WebData:1 ]
WebFS   (ocf::heartbeat:Filesystem):    Started pcmk-2
```

Once we’ve done everything we needed to on pcmk-1 (in this case nothing, we just wanted to see the resources move), we can allow the node to be a full cluster member again.

```
[root@pcmk-1 ~]# crm node online
[root@pcmk-1 ~]# crm_mon
============
Last updated: Tue Sep  1 10:13:25 2009
Stack: openais
```

```
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
4 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
WebSite (ocf::heartbeat:apache):        Started pcmk-2
Master/Slave Set: WebDataClone
        Masters: [ pcmk-2 ]
        Slaves: [ pcmk-1 ]
WebFS   (ocf::heartbeat:Filesystem):    Started pcmk-2
```

Notice that our resource stickiness settings prevent the services from migrating back to pcmk-1.

Conversion to Active/Active

8.1. Requirements

The primary requirement for an Active/Active cluster is that the data required for your services are available, simultaneously, on both machines. Pacemaker makes no requirement on how this is achieved, you could use a SAN if you had one available, however since DRBD supports multiple Primaries, we can also use that.

The only hitch is that we need to use a cluster-aware filesystem (and the one we used earlier with DRBD, ext4, is not one of those). Both OCFS2 and GFS2 are supported, however here we will use GFS2 which comes with Fedora 13 .

8.2. Install a Cluster Filesystem - GFS2

The first thing to do is install gfs2-utils on each machine.

```
[root@pcmk-1 ~]# yum install -y gfs2-utils gfs-pcmk
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package gfs-pcmk.x86_64 0:3.0.5-2.fc12 set to be updated
--> Processing Dependency: libSaCkpt.so.3(OPENAIS_CKPT_B.01.01)(64bit) for package: gfs-
pcmk-3.0.5-2.fc12.x86_64
--> Processing Dependency: dlm-pcmk for package: gfs-pcmk-3.0.5-2.fc12.x86_64
--> Processing Dependency: libccs.so.3()(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
--> Processing Dependency: libdlmcontrol.so.3()(64bit) for package: gfs-
pcmk-3.0.5-2.fc12.x86_64
--> Processing Dependency: liblogthread.so.3()(64bit) for package: gfs-
pcmk-3.0.5-2.fc12.x86_64
--> Processing Dependency: libSaCkpt.so.3()(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
---> Package gfs2-utils.x86_64 0:3.0.5-2.fc12 set to be updated
--> Running transaction check
---> Package clusterlib.x86_64 0:3.0.5-2.fc12 set to be updated
---> Package dlm-pcmk.x86_64 0:3.0.5-2.fc12 set to be updated
---> Package openaislib.x86_64 0:1.1.0-1.fc12 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch               Version                   Repository        Size
================================================================================
Installing:
 gfs-pcmk               x86_64             3.0.5-2.fc12              custom           101 k
 gfs2-utils             x86_64             3.0.5-2.fc12              custom           208 k
Installing for dependencies:
 clusterlib             x86_64             3.0.5-2.fc12              custom            65 k
 dlm-pcmk               x86_64             3.0.5-2.fc12              custom            93 k
 openaislib             x86_64             1.1.0-1.fc12              fedora            76 k

Transaction Summary
================================================================================
Install       5 Package(s)
Upgrade       0 Package(s)

Total download size: 541 k
```

```
Downloading Packages:
(1/5): clusterlib-3.0.5-2.fc12.x86_64.rpm                                  |  65 kB     00:00
(2/5): dlm-pcmk-3.0.5-2.fc12.x86_64.rpm                                    |  93 kB     00:00
(3/5): gfs-pcmk-3.0.5-2.fc12.x86_64.rpm                                    | 101 kB     00:00
(4/5): gfs2-utils-3.0.5-2.fc12.x86_64.rpm                                  | 208 kB     00:00
(5/5): openaislib-1.1.0-1.fc12.x86_64.rpm                                  |  76 kB     00:00
--------------------------------------------------------------------------------------------
Total                                                       992 kB/s | 541 kB     00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : clusterlib-3.0.5-2.fc12.x86_64                                     1/5
  Installing     : openaislib-1.1.0-1.fc12.x86_64                                     2/5
  Installing     : dlm-pcmk-3.0.5-2.fc12.x86_64                                       3/5
  Installing     : gfs-pcmk-3.0.5-2.fc12.x86_64                                       4/5
  Installing     : gfs2-utils-3.0.5-2.fc12.x86_64                                     5/5

Installed:
  gfs-pcmk.x86_64 0:3.0.5-2.fc12                    gfs2-utils.x86_64 0:3.0.5-2.fc12

Dependency Installed:
  clusterlib.x86_64 0:3.0.5-2.fc12   dlm-pcmk.x86_64 0:3.0.5-2.fc12
  openaislib.x86_64 0:1.1.0-1.fc12

Complete!
[root@pcmk-1 x86_64]#
```

8.3. Setup Pacemaker-GFS2 Integration

GFS2 needs two services to be running, the first is the user-space interface to the kernel's distributed lock manager (DLM). The DLM is used to co-ordinate which node(s) can access a given file (and when) and integrates with Pacemaker to obtain node membership [1] information and fencing capabilities.

The second service is GFS2's own control daemon which also integrates with Pacemaker to obtain node membership data.

8.3.1. Add the DLM service

The DLM control daemon needs to run on all active cluster nodes, so we will use the shells interactive mode to create a cloned resource.

```
[root@pcmk-1 ~]# crm
crm(live)# cib new stack-glue
INFO: stack-glue shadow CIB created
crm(stack-glue)# configure primitive dlm ocf:pacemaker:controld op monitor interval=120s
crm(stack-glue)# configure clone dlm-clone dlm meta interleave=true
crm(stack-glue)# configure show xml
crm(stack-glue)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
```

[1] The list of nodes the cluster considers to be available

```
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="ext4"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
 op monitor interval="120s"
ms WebDataClone WebData \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone dlm-clone dlm \
 meta interleave="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

Note

TODO: Explain the meaning of the interleave option

Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster’s response

```
crm(stack-glue)# cib commit stack-glue
INFO: commited 'stack-glue' shadow CIB to the cluster
crm(stack-glue)# quit
bye
[root@pcmk-1 ~]# crm_mon
============
Last updated: Thu Sep  3 20:49:54 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
5 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

WebSite (ocf::heartbeat:apache):        Started pcmk-2
Master/Slave Set: WebDataClone
        Masters: [ pcmk-1 ]
        Slaves: [ pcmk-2 ]
ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
Clone Set: dlm-clone
```

```
 Started: [ pcmk-2 pcmk-1 ]
WebFS   (ocf::heartbeat:Filesystem):    Started pcmk-2
```

8.3.2. Add the GFS2 service

Once the DLM is active, we can add the GFS2 control daemon.

Use the crm shell to create the gfs-control cluster resource:

```
[root@pcmk-1 ~]# crm
crm(live)# cib new gfs-glue --force
INFO: gfs-glue shadow CIB created
crm(gfs-glue)# configure primitive gfs-control ocf:pacemaker:controld params
 daemon=gfs_controld.pcmk args="-g 0" op monitor interval=120s
crm(gfs-glue)# configure clone gfs-clone gfs-control meta interleave=true
```

Now ensure Pacemaker only starts the gfs-control service on nodes that also have a copy of the dlm service (created above) already running

```
crm(gfs-glue)# configure colocation gfs-with-dlm INFINITY: gfs-clone dlm-clone
crm(gfs-glue)# configure order start-gfs-after-dlm mandatory: dlm-clone gfs-clone
```

Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster's response

```
crm(gfs-glue)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="ext4"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
        op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \
 params daemon="gfs_controld.pcmk" args="-g 0" \
 op monitor interval="120s"
ms WebDataClone WebData \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone dlm-clone dlm \
        meta interleave="true"
clone gfs-clone gfs-control \
 meta interleave="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
```

```
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
crm(gfs-glue)# cib commit gfs-glue
INFO: commited 'gfs-glue' shadow CIB to the cluster
crm(gfs-glue)# quit
bye
[root@pcmk-1 ~]# crm_mon
============
Last updated: Thu Sep  3 20:49:54 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

WebSite (ocf::heartbeat:apache):        Started pcmk-2
Master/Slave Set: WebDataClone
        Masters: [ pcmk-1 ]
        Slaves: [ pcmk-2 ]
ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
Clone Set: dlm-clone
        Started: [ pcmk-2 pcmk-1 ]
Clone Set: gfs-clone
 Started: [ pcmk-2 pcmk-1 ]
WebFS   (ocf::heartbeat:Filesystem):    Started pcmk-1
```

8.4. Create a GFS2 Filesystem

8.4.1. Preparation

Before we do anything to the existing partition, we need to make sure it is unmounted. We do this by tell the cluster to stop the WebFS resource. This will ensure that other resources (in our case, Apache) using WebFS are not only stopped, but stopped in the correct order.

```
[root@pcmk-1 ~]# crm_resource --resource WebFS --set-parameter target-role --meta --parameter-
value Stopped
[root@pcmk-1 ~]# crm_mon
============
Last updated: Thu Sep  3 15:18:06 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]
```

```
Master/Slave Set: WebDataClone
        Masters: [ pcmk-1 ]
        Slaves: [ pcmk-2 ]
ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-1
Clone Set: dlm-clone
        Started: [ pcmk-2 pcmk-1 ]
Clone Set: gfs-clone
        Started: [ pcmk-2 pcmk-1 ]
```

Note

Note that both Apache and WebFS have been stopped.

8.4.2. Create and Populate an GFS2 Partition

Now that the cluster stack and integration pieces are running smoothly, we can create an GFS2 partition.

Warning

This will erase all previous content stored on the DRBD device. Ensure you have a copy of any important data.

We need to specify a number of additional parameters when creating a GFS2 partition.

First we must use the -p option to specify that we want to use the the Kernel's DLM. Next we use -j to indicate that it should reserve enough space for two journals (one per node accessing the filesystem).

Lastly, we use -t to specify the lock table name. The format for this field is clustername:fsname. For the fsname, we just need to pick something unique and descriptive and since we haven't specified a clustername yet, we will use the default (pcmk).

To specify an alternate name for the cluster, locate the service section containing "name: pacemaker" in corosync.conf and insert the following line anywhere inside the block:

clustername: myname

Do this on each node in the cluster and be sure to restart them before continuing.

```
mkfs.gfs2 -p lock_dlm -j 2 -t pcmk:web /dev/drbd1
[root@pcmk-1 ~]# mkfs.gfs2 -t pcmk:web -p lock_dlm -j 2 /dev/vdb
This will destroy any data on /dev/vdb.
It appears to contain: data

Are you sure you want to proceed? [y/n] y

Device:                    /dev/vdb
Blocksize:                 4096
Device Size                1.00 GB (131072 blocks)
Filesystem Size:           1.00 GB (131070 blocks)
Journals:                  2
Resource Groups:           2
```

```
Locking Protocol:          "lock_dlm"
Lock Table:                "pcmk:web"
UUID:                      6B776F46-177B-BAF8-2C2B-292C0E078613

[root@pcmk-1 ~]#
```

Then (re)populate the new filesystem with data (web pages). For now we'll create another variation on our home page.

```
[root@pcmk-1 ~]# mount /dev/drbd1 /mnt/
[root@pcmk-1 ~]# cat <<-END >/mnt/index.html
<html>
<body>My Test Site - GFS2</body>
</html>
END
[root@pcmk-1 ~]# umount /dev/drbd1
[root@pcmk-1 ~]# drbdadm verify wwwdata
[root@pcmk-1 ~]#
```

8.5. Reconfigure the Cluster for GFS2

```
[root@pcmk-1 ~]# crm
crm(live)# cib new GFS2
INFO: GFS2 shadow CIB created
crm(GFS2)# configure delete WebFS
crm(GFS2)# configure primitive WebFS ocf:heartbeat:Filesystem params device="/dev/drbd/by-res/
wwwdata" directory="/var/www/html" fstype="gfs2"
```

Now that we've recreated the resource, we also need to recreate all the constraints that used it. This is because the shell will automatically remove any constraints that referenced WebFS.

```
crm(GFS2)# configure colocation WebSite-with-WebFS inf: WebSite WebFS
crm(GFS2)# configure colocation fs_on_drbd inf: WebFS WebDataClone:Master
crm(GFS2)# configure order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
crm(GFS2)# configure order WebSite-after-WebFS inf: WebFS WebSite
crm(GFS2)# configure colocation WebFS-with-gfs-control INFINITY: WebFS gfs-clone
crm(GFS2)# configure order start-WebFS-after-gfs-control mandatory: gfs-clone WebFS
crm(GFS2)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
 params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="gfs2"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
        op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \
   params daemon="gfs_controld.pcmk" args="-g 0" \
        op monitor interval="120s"
```

```
ms WebDataClone WebData \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone dlm-clone dlm \
        meta interleave="true"
clone gfs-clone gfs-control \
        meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFS gfs-clone
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
order start-WebFS-after-gfs-control inf: gfs-clone WebFS
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster's response

```
crm(GFS2)# cib commit GFS2
INFO: commited 'GFS2' shadow CIB to the cluster
crm(GFS2)# quit
bye
[root@pcmk-1 ~]# crm_mon
============
Last updated: Thu Sep  3 20:49:54 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

WebSite (ocf::heartbeat:apache):        Started pcmk-2
Master/Slave Set: WebDataClone
        Masters: [ pcmk-1 ]
        Slaves: [ pcmk-2 ]
ClusterIP        (ocf::heartbeat:IPaddr):        Started pcmk-2
Clone Set: dlm-clone
        Started: [ pcmk-2 pcmk-1 ]
Clone Set: gfs-clone
        Started: [ pcmk-2 pcmk-1 ]
WebFS (ocf::heartbeat:Filesystem): Started pcmk-1
```

8.6. Reconfigure Pacemaker for Active/Active

Almost everything is in place. Recent versions of DRBD are capable of operating in Primary/Primary mode and the filesystem we're using is cluster aware. All we need to do now is reconfigure the cluster to take advantage of this.

This will involve a number of changes, so we'll again use interactive mode.

```
[root@pcmk-1 ~]# crm
[root@pcmk-1 ~]# cib new active
```

There's no point making the services active on both locations if we can't reach them, so lets first clone the IP address. Cloned IPaddr2 resources use an iptables rule to ensure that each request only processed by one of the two clone instances. The additional meta options tell the cluster how many instances of the clone we want (one "request bucket" for each node) and that if all other nodes fail, then the remaining node should hold all of them. Otherwise the requests would be simply discarded.

```
[root@pcmk-1 ~]# configure clone WebIP ClusterIP \
 meta globally-unique="true" clone-max="2" clone-node-max="2"
```

Now we must tell the ClusterIP how to decide which requests are processed by which hosts. To do this we must specify the clusterip_hash parameter.

Open the ClusterIP resource

```
[root@pcmk-1 ~]# configure edit ClusterIP
```

And add the following to the params line

```
clusterip_hash="sourceip"
```

So that the complete definition looks like:

```
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
        op monitor interval="30s"
```

Here is the full transcript

```
[root@pcmk-1 ~]# crm
crm(live)# cib new active
INFO: active shadow CIB created
crm(active)# configure clone WebIP ClusterIP \
        meta globally-unique="true" clone-max="2" clone-node-max="2"
crm(active)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="gfs2"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
```

```
        op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \
   params daemon=”gfs_controld.pcmk” args=”-g 0” \
        op monitor interval="120s"
ms WebDataClone WebData \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone WebIP ClusterIP \
 meta globally-unique=”true” clone-max=”2” clone-node-max=”2”
clone dlm-clone dlm \
        meta interleave="true"
clone gfs-clone gfs-control \
        meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFS gfs-clone
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSite WebIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: WebIP WebSite
order start-WebFS-after-gfs-control inf: gfs-clone WebFS
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

Notice how any constraints that referenced ClusterIP have been updated to use WebIP instead. This is an additional benefit of using the crm shell.

Next we need to convert the filesystem and Apache resources into clones. Again, the shell will automatically update any relevant constraints.

```
crm(active)# configure clone WebFSClone WebFS
crm(active)# configure clone WebSiteClone WebSite
```

The last step is to tell the cluster that it is now allowed to promote both instances to be Primary (aka. Master).

```
crm(active)# configure edit WebDataClone
```

Change master-max to 2

```
crm(active)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype=”gfs2”
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
```

```
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip=”192.168.122.101” cidr_netmask=”32” clusterip_hash=”sourceip” \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
        op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \
   params daemon=”gfs_controld.pcmk” args=”-g 0” \
        op monitor interval="120s"
ms WebDataClone WebData \
        meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone WebFSClone WebFS
clone WebIP ClusterIP  \
        meta globally-unique=”true” clone-max=”2” clone-node-max=”2”
clone WebSiteClone WebSite
clone dlm-clone dlm \
        meta interleave="true"
clone gfs-clone gfs-control \
        meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSiteClone WebIP
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
order apache-after-ip inf: WebIP WebSiteClone
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="false" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster’s response

```
crm(active)# cib commit active
INFO: commited 'active' shadow CIB to the cluster
crm(active)# quit
bye
[root@pcmk-1 ~]# crm_mon
============
Last updated: Thu Sep  3 21:37:27 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

Master/Slave Set: WebDataClone
        Masters: [ pcmk-1 pcmk-2 ]
Clone Set: dlm-clone
        Started: [ pcmk-2 pcmk-1 ]
Clone Set: gfs-clone
```

```
        Started: [ pcmk-2 pcmk-1 ]
Clone Set: WebIP
 Started: [ pcmk-1 pcmk-2 ]
Clone Set: WebFSClone
 Started: [ pcmk-1 pcmk-2 ]
Clone Set: WebSiteClone
 Started: [ pcmk-1 pcmk-2 ]
```

8.6.1. Testing Recovery

Note

TODO: Put one node into standby to demonstrate failover

Configure STONITH

9.1. Why You Need STONITH

STONITH is an acronym for Shoot-The-Other-Node-In-The-Head and it protects your data from being corrupted by rouge nodes or concurrent access.

Just because a node is unresponsive, this doesn't mean it isn't accessing your data. The only way to be 100% sure that your data is safe, is to use STONITH so we can be certain that the node is truly offline, before allowing the data to be accessed from another node.

STONITH also has a role to play in the event that a clustered service cannot be stopped. In this case, the cluster uses STONITH to force the whole node offline, thereby making it safe to start the service elsewhere.

9.2. What STONITH Device Should You Use

It is crucial that the STONITH device can allow the cluster to differentiate between a node failure and a network one.

The biggest mistake people make in choosing a STONITH device is to use remote power switch (such as many onboard IMPI controllers) that shares power with the node it controls. In such cases, the cluster cannot be sure if the node is really offline, or active and suffering from a network fault.

Likewise, any device that relies on the machine being active (such as SSH-based "devices" used during testing) are inappropriate.

9.3. Configuring STONITH

1. Find the correct driver: stonith -L
2. Since every device is different, the parameters needed to configure it will vary. To find out the parameters required by the device: stonith -t {type} -n

Hopefully the developers chose names that make sense, if not you can query for some additional information by finding an active cluster node and running:

```
lrmadmin -M stonith {type} pacemaker
```

The output should be XML formatted text containing additional parameter descriptions

1. Create a file called stonith.xml containing a primitive resource with a class of stonith, a type of {type} and a parameter for each of the values returned in step 2
2. Create a clone from the primitive resource if the device can shoot more than one node *and supports multiple simultaneous connections*.
3. Upload it into the CIB using cibadmin: cibadmin -C -o resources --xml-file stonith.xml

9.3.1. Example

Assuming we have an IBM BladeCenter containing our two nodes and the management interface is active on 192.168.122.31, then we would chose the external/ibmrsa driver in step 2 and obtain the following list of parameters

```
stonith -t external/ibmrsa -n
[root@pcmk-1 ~]# stonith -t external/ibmrsa -n
hostname  ipaddr  userid  passwd  type
```

Assuming we know the username and password for the management interface, we would create a STONITH resource with the shell

```
[root@pcmk-1 ~]# crm
crm(live)# cib new stonith
INFO: stonith shadow CIB created
crm(stonith)# configure primitive rsa-fencing stonith::external/ibmrsa \
 params hostname="pcmk-1 pcmk-2" ipaddr=192.168.122.31 userid=mgmt passwd=abc123 type=ibm \
 op monitor interval="60s"
crm(stonith)# configure clone Fencing rsa-fencing
```

And finally, since we disabled it earlier, we need to re-enable STONITH

```
crm(stonith)# configure property stonith-enabled="true"
crm(stonith)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="gfs2"
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
        op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \
   params daemon="gfs_controld.pcmk" args="-g 0" \
        op monitor interval="120s"
primitive rsa-fencing stonith::external/ibmrsa \
 params hostname="pcmk-1 pcmk-2" ipaddr=192.168.122.31 userid=mgmt passwd=abc123 type=ibm \
 op monitor interval="60s"
ms WebDataClone WebData \
        meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone Fencing rsa-fencing
clone WebFSClone WebFS
clone WebIP ClusterIP  \
        meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSite
clone dlm-clone dlm \
        meta interleave="true"
clone gfs-clone gfs-control \
        meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
```

```
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSiteClone WebIP
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
order apache-after-ip inf: WebIP WebSiteClone
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled="true" \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

Appendix A. Configuration Recap

A.1. Final Cluster Configuration

```
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype=”gfs2”
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip=”192.168.122.101” cidr_netmask=”32” clusterip_hash=”sourceip” \
        op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \
        op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \
   params daemon=”gfs_controld.pcmk” args=”-g 0” \
        op monitor interval="120s"
primitive rsa-fencing stonith::external/ibmrsa \
        params hostname=”pcmk-1 pcmk-2" ipaddr=192.168.122.31 userid=mgmt passwd=abc123
 type=ibm \
        op monitor interval="60s"
ms WebDataClone WebData \
        meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone Fencing rsa-fencing
clone WebFSClone WebFS
clone WebIP ClusterIP  \
        meta globally-unique=”true” clone-max=”2” clone-node-max=”2”
clone WebSiteClone WebSite
clone dlm-clone dlm \
        meta interleave="true"
clone gfs-clone gfs-control \
        meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSiteClone WebIP
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
order apache-after-ip inf: WebIP WebSiteClone
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled=”true” \
        no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

A.2. Node List

The list of cluster nodes is automatically populated by the cluster.

```
node pcmk-1
node pcmk-2
```

A.3. Cluster Options

This is where the cluster automatically stores some information about the cluster

1. dc-version - the version (including upstream source-code hash) of Pacemaker used on the DC
2. cluster-infrastructure - the cluster infrastructure being used (heartbeat or openais)
3. expected-quorum-votes - the maximum number of nodes expected to be part of the cluster

and where the admin can set options that control the way the cluster operates

1. stonith-enabled=true - Make use of STONITH
2. no-quorum-policy=ignore - Ignore loss of quorum and continue to host resources.

```
property $id="cib-bootstrap-options" \
        dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
        cluster-infrastructure="openais" \
        expected-quorum-votes=”2” \
        stonith-enabled=”true” \
        no-quorum-policy="ignore"
```

A.4. Resources

A.4.1. Default Options

Here we configure cluster options that apply to every resource.

1. resource-stickiness - Specify the aversion to moving resources to other machines

```
rsc_defaults $id="rsc-options" \
        resource-stickiness=”100”
```

A.4.2. Fencing

Note

TODO: Add text here

```
primitive rsa-fencing stonith::external/ibmrsa \
        params hostname=”pcmk-1 pcmk-2" ipaddr=192.168.122.31 userid=mgmt passwd=abc123
 type=ibm \
        op monitor interval="60s"
clone Fencing rsa-fencing
```

A.4.3. Service Address

Users of the services provided by the cluster require an unchanging address with which to access it. Additionally, we cloned the address so it will be active on both nodes. An iptables rule (created as part of the resource agent) is used to ensure that each request only processed by one of the two clone instances. The additional meta options tell the cluster that we want two instances of the clone (one “request bucket” for each node) and that if one node fails, then the remaining node should hold both.

```
primitive ClusterIP ocf:heartbeat:IPaddr2 \
        params ip=”192.168.122.101” cidr_netmask=”32” clusterip_hash=”sourceip” \
        op monitor interval="30s"
clone WebIP ClusterIP
        meta globally-unique=”true” clone-max=”2” clone-node-max=”2”
```

Note

TODO: The RA should check for globally-unique=true when cloned

A.4.4. Distributed lock manager

Cluster filesystems like GFS2 require a lock manager. This service starts the daemon that provides user-space applications (such as the GFS2 daemon) with access to the in-kernel lock manager. Since we need it to be available on all nodes in the cluster, we have it cloned.

```
primitive dlm ocf:pacemaker:controld \
        op monitor interval="120s"
clone dlm-clone dlm \
        meta interleave="true
```

Note

TODO: Confirm **`interleave`** is no longer needed

A.4.5. GFS control daemon

GFS2 also needs a user-space/kernel bridge that runs on every node. So here we have another clone, however this time we must also specify that it can only run on machines that are also running the DLM (colocation constraint) and that it can only be started after the DLM is running (order constraint). Additionally, the gfs-control clone should only care about the DLM instances it is paired with, so we need to set the interleave option.

```
primitive gfs-control ocf:pacemaker:controld \
   params daemon=”gfs_controld.pcmk” args=”-g 0” \
        op monitor interval="120s"
clone gfs-clone gfs-control \
        meta interleave="true"
colocation gfs-with-dlm inf: gfs-clone dlm-clone
order start-gfs-after-dlm inf: dlm-clone gfs-clone
```

A.4.6. DRBD - Shared Storage

Here we define the DRBD service and specify which DRBD resource (from drbd.conf) it should manage. We make it a master/slave resource and, in order to have an active/active setup, allow both instances to be promoted by specifying master-max=2. We also set the notify option so that the cluster will tell DRBD agent when it’s peer changes state.

```
primitive WebData ocf:linbit:drbd \
        params drbd_resource="wwwdata" \
        op monitor interval="60s"
ms WebDataClone WebData \
        meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
```

A.4.7. Cluster Filesystem

The cluster filesystem ensures that files are read and written correctly. We need to specify the block device (provided by DRBD), where we want it mounted and that we are using GFS2. Again it is a clone because it is intended to be active on both nodes. The additional constraints ensure that it can only be started on nodes with active gfs-control and drbd instances.

```
primitive WebFS ocf:heartbeat:Filesystem \
        params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype=”gfs2”
clone WebFSClone WebFS
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
```

A.4.8. Apache

Lastly we have the actual service, Apache. We need only tell the cluster where to find it’s main configuration file and restrict it to running on nodes that have the required filesystem mounted and the IP address active.

```
primitive WebSite ocf:heartbeat:apache \
        params configfile="/etc/httpd/conf/httpd.conf" \
        op monitor interval="1min"
clone WebSiteClone WebSite
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation website-with-ip inf: WebSiteClone WebIP
order apache-after-ip inf: WebIP WebSiteClone
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
```

Appendix B. Sample Corosync.conf

```
# Please read the Corosync.conf.5 manual page
compatibility: whitetank

aisexec {
        # Run as root - this is necessary to be able to manage resources with Pacemaker
        user:        root
        group:        root
}

service {
        # Load the Pacemaker Cluster Resource Manager
        ver:       0
        name:      pacemaker
        use_mgmtd: no
        use_logd:  no
}

totem {
        version: 2

        # How long before declaring a token lost (ms)
        token:          5000

        # How many token retransmits before forming a new configuration
        token_retransmits_before_loss_const: 10

        # How long to wait for join messages in the membership protocol (ms)
        join:           1000

        # How long to wait for consensus to be achieved before starting a new
        # round of membership configuration (ms)
        consensus:      2500

        # Turn off the virtual synchrony filter
        vsftype:        none

        # Number of messages that may be sent by one processor on receipt of the token
        max_messages:   20

        # Stagger sending the node join messages by 1..send_join ms
        send_join: 45

        # Limit generated nodeids to 31-bits (positive signed integers)
        clear_node_high_bit: yes

        # Disable encryption
        secauth:        off

        # How many threads to use for encryption/decryption
        threads:           0

        # Optionally assign a fixed node id (integer)
        # nodeid:         1234

        interface {
                ringnumber: 0

                # The following values need to be set based on your environment
                bindnetaddr: 192.168.122.0
```

```
                mcastaddr: 226.94.1.1
                mcastport: 4000
        }
}

logging {
        debug: off
        fileline: off
        to_syslog: yes
        to_stderr: off
        syslog_facility: daemon
        timestamp: on
}

amf {
        mode: disabled
}
```

Appendix C. Further Reading

Project Website

http://www.clusterlabs.org[1]

Cluster Commands

A comprehensive guide to cluster commands has been written by Novell and can be found at:

http://www.novell.com/documentation/sles11/book_sleha/index.html?page=/documentation/sles11/book_sleha/data/book_sleha.html

Corosync

http://www.corosync.org[2]

[1] http://www.clusterlabs.org/
[2] http://www.corosync.org/

Appendix D. Revision History

Revision 1 Mon May 17 2010 Andrew Beekhof *andrew@beekhof.net*

Import from Pages.app

Index

F